Printing Requisition/Graphic Services

Dept. No. 20878

#1265 369.

Title or Description
Des. & Imp. Of Pkg. for SSS of Ord. Dif. Eqs.(Research Rep. CS-12-79)

Date 17 Dec. ~~12 March~~ 1979 KOG
Date Required
Account 126-6242-41

Signature Virginia/Edith Huang
Signing Authority K.O. Geddes

Department Computer Science
Room 5100B
Phone 3402
Delivery ☐ Mail ☐ Pick-up ☐ Via Stores ☐ Other

1. Please complete unshaded areas on form as applicable. (4 part no carbon required).
2. Distribute copies as follows: White, Canary and Pink—Printing, Arts Library or applicable Copy Centre Goldenrod—**Retain.**
3. On completion of order, pink copy will be returned with printed material. Canary copy will be costed and returned to requisitioner, **Retain as a record of your charges.**
4. Please direct enquiries, quoting requisition number, to Printing/Graphic Services, Extension 3451.

Reproduction Requirements ☒ Offset ☐ Signs/Repro's ☐ Xerox
Number of Pages 120 ++
Number of Copies 50

Type of Paper Stock ☒ Bond ☐ Book ☐ Cover ☐ Bristol ☑ Supplied COVERS & BACKS

Paper Size ☒ 8½ x 11 ☐ 8½ x 14 ☐ 11 x 17

Paper Colour ☒ White ☐ Other
Ink ☐ Black

Printing ☒ 1 Side ☐ 2 Sides
Numbering to

Binding/Finishing Operations ☒ Collating ☐ Corner Stitching ☐ 3 Ring ☐ Tape ☐ Plastic Ring ☐ Perforating

Folding Finished Size 3 STAPLES
Cutting Finished Size

Special Instructions
Please enclose in front and back covers supplied and staple with three staples down the left-hand side.

Copy Time/Materials — Signs/Repro's — Camera — Collecting & Masking Negatives — Platemaking — Printing — Bindery

Film Qty Size — Plates Qty Size & Type — Paper Qty Size — Binder/Rings Qty Size — Outside Services

Graphic Services May 68 482-2

The Design and Implementation of a Package for Symbolic Series Solution of Ordinary Differential Equations.

by

M.O. Afolabi[†]

Computer Science Dept.
Research Report CS-79-12

University of Waterloo
Waterloo, Ontario, CANADA

March 1979

[†]Present address: Department of Computer Sciences
University of Lagos
Lagos
NIGERIA

The Design and Implementation of a Package for Symbolic Series Solution of Ordinary Differential Equations

Abstract

This paper describes the design and implementation of SODEPACK, the Symbolic Ordinary Differential Equation Package, for computing a differential equation (ODE). The approximation is expressed in terms of the indeterminates (if any) present in the ODE or its associated conditions. A very simple user/package interface is provided which enables the user to specify his problem in a natural form. The output from the package is in a clear, readable form.

1. Introduction

This paper describes the structure, design, and implementation of SODEPACK, the Symbolic Ordinary Differential Equation Package, which is designed to compute a polynomial approximation of given degree to a ν-order ODE of the form

$$G(x,y(x),y'(x),\ldots,y^{(\nu)}(x)) = 0,$$

where G is a polynomial function in $x,y(x),y'(x),\ldots,y^{(\nu)}(x)$. The ν conditions associated with the ODE may be quite arbitrary. In general, any linear combination of function and derivative (up to order ν-1) values at one or more points is allowed. The polynomial approximation is expressed in terms of the indeterminates (if any) present in the ODE or its associated conditions.

Programs for symbolic computation usually involve some fairly complex data and storage structures. A user of these programs would like to specify his problem in a 'natural' form and not be concerned with how the problem is 'massaged' to conform to a format or structure required by a particular programming language and/or computer installation. The output must also be in a 'human-readable' form which reflects the nature of the problem being solved. Even in a package for numerical computation these features are highly desirable, (see, for example, [3,4]).

SODEPACK insulates the user from most system considerations, accepting the ODE to be solved (together with the associated conditions) in a reasonably natural form. The output is also in a clear, readable and adequately documented form. The package provides a variety of

methods for solving ODE's and we offer suggestions for selecting a method for a given problem.

But perhaps the most attractive feature of the package is the simplicity of the user/package interface. The main programs calling the routines in the package are very simple, consisting of less than ten ALTRAN [2] statements, for most problems. The package may be used to solve several problems using a particular method, in one run.

In Section 2 we shall describe the basic structure of SODEPACK. A more detailed description may be found in [1], which also contains examples of mainline programs. In Section 3 we give the listings of the procedures in the package.

2. Basic Structure of the Package

SODEPACK offers a collection of methods for solving general ν-order ODE's of the form given in Section 1. For all the methods in the package, the user and the package interact to solve the ODE problem through the following simple steps:

Step 1: The user calls the module initialization procedure for a particular method, thereby defining the structure of G and the associated conditions.

Step 2: The user calls the module for the appropriate method (see Figure 1). The package reads in the problem and outputs the polynomial approximation of required degree in both the power and Chebyshev series forms, together with timing statistics and error estimates (where applicable).

The different methods provided in the package correspond to different algorithms for obtaining the polynomial approximation for linear and nonlinear ODE's. The general structure of the methods used in the package may be represented graphically by Figure 1.

On the whole, there are four methods and the user chooses a particular method by calling the appropriate routines in Steps 1 and 2 above. The methods are distinguished by the last two characters of the procedure names as follows (YYYY indicates the function performed by the procedure):

YYYYXi

X i

C 1 Chebyshev Series Method for General Linear ODE

T 1 TAU Method for General LODE Via Series Substitution

T 2 TAU Method for General LODE Via Canonical Polynomials

N 1 TAU Method for General Nonlinear ODE Via Series Substitution

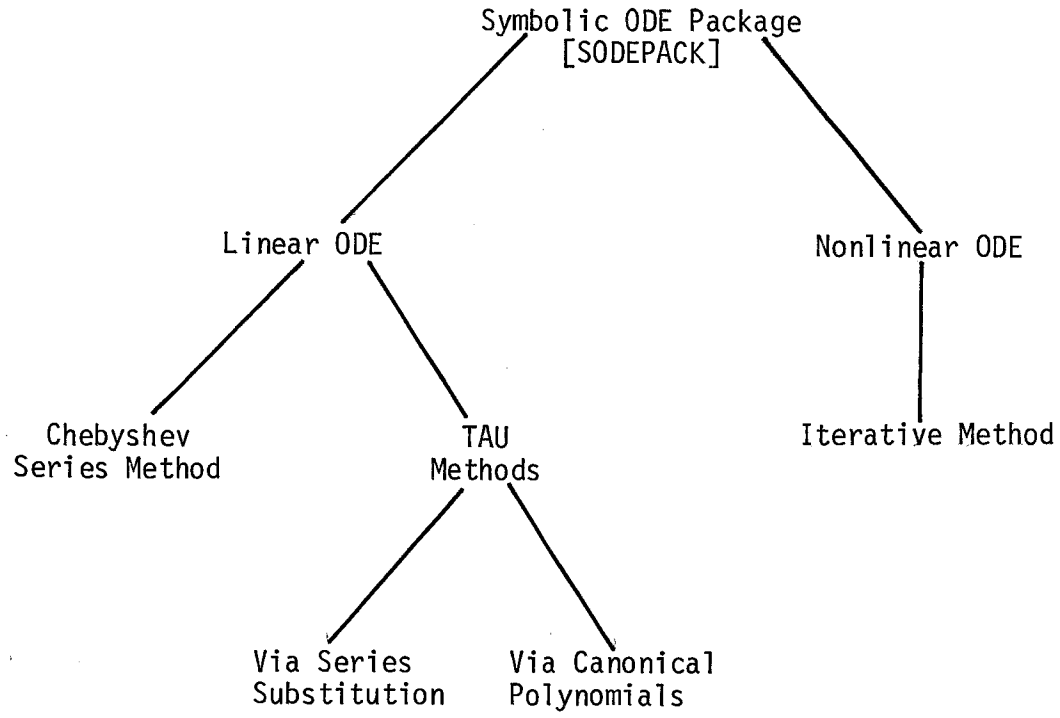

Figure 1 Overall Structure of Methods in SODEPACK

The procedures which make up SODEPACK are divided into six levels as given in Figure 2.

LEVEL					
1	2	3	4	5	6
				INPCON	INFORM
PINTXi				INPODE	POWERS
				PRODTK	TPOWS
SODEXi	PROX			INT	EVAL
	SOLXi	RECXij		INIT	TRUNC
		GSXij	PSXi	CANONP	CHFORM
			VALXij	SETP1	TCS
				SETP2	ABS
				INTEGR	LQABS
				ERREST	AROUND
				etc.	QROUND
	EPIX				etc.

Figure 2. Overall Structure of SODEPACK

Level 1 These procedures form the user/package interface. PINTXi are the package initialization routines for the different methods, while SODEXi are the actual implementations of the methods.

Level 2 This level consists of the procedures called by SODEXi, namely the prologue, PROX, for inputting and checking data, SOLXi for solving the problem, and EPIX for output requirements.

Level 3 This level consists of procedures used by SOLXi to generate the solution required. RECXij are the routines for symbolic generation of the relevant recurrence equations. The procedures GSXi compute the required general solution of the recurrence equations.

Level 4 This level consists of the different procedures used by RECXij and GSXij corresponding to the form of the ODE and the solution codes (see [1, Section 5.6]) specified for the problem. The procedures PSXij compute the parametric solutions of the problem and VALXij assign values to the parameters present in these solutions.

Level 5 This level consists of the package utility procedures, that is, procedures called by any of the procedures in levels 1-4. The use of these procedures is restricted to SODEPACK. Examples are INPODE and INPCON which read in a differential equation and its associated conditions, respectively.

Level 6 This level consists of the general utility procedures, i.e. procedures called by any of the procedures in levels 1-5 above. These are very basic procedures that could very well form part of a general system function library. Examples are the absolute value routines ABS (real), LQABS (long rational), and Chebyshev series evaluation routine EVAL, for Clenshaw's rule.

Although all the levels are shown in Figure 2, we point out again that only the procedures in Level 1 are visible to the user.

Restrictions and Asssumptions

(1) The package assumes that the solution is required in a finite interval [a,b]. However, the indeterminates appearing in the ODE are in the interval [-1,1].

(2) The order of the ODE to be solved may not exceed 5.

(3) For linear ODE's the degree of approximation may not exceed 20; for nonlinear ODE's the degree may not exceed 15.

(4) The degree of the ODE may not exceed 4.

3. Listings of Procedures

In this section we give the listings of the procedures SODEPACK. The procedures are described in more detail in [1].

~~APPENDIX B~~

~~LISTINGS OF PROCEDURES~~

3. Listings of Procedures

In this section we ~~This appendix~~ gives the listings of the procedures ~~in Levels 1, 2 and 3 of~~ SODEPACK ~~(see Figure 5.3 of Section 5.2)~~. The procedures are described in more detail in ~~Section 5.3.~~ [1]. ~~The complete listings of all the procedures in the package may be found in [Afo79].~~

List of Level 1 Procedures

List of Level 1 Procedures

*Not a Level 1 Procedure, but included here because it performs most of the work required in PINTXi.

```
PROCEDURE PINTC1
   INTEGER ARRAY DEFLT
#  THIS PROCEDURE SETS THE DEFAULT VALUES FOR SOLVER MODULE [illegible]
#   SOPLIT FOR THE VARIABLES MEMORD,MAXDEG,MAXVAL,MAXMIT,MAXCON
#   AND CALLS PROCEDURE PKINIT TO PERFORM THE ACTUAL
#   INITIALIZATION.
#  PACKAGE PROCEDURE REQUIRED: PKINIT.
#  DECLARATION IN CALLING PROCEDURE: NONE.

   DEFLT = (2,2,5,1,2);
   PKINIT(DEFLT);

END  #  END OF PROCEDURE PINTC1.
```

```
PROCEDURE PINTT1
   INTEGER ARRAY DEFLT
#  THIS PROCEDURE SETS THE DEFAULT VALUES FOR SOLVER MODULE
#   SODET1 FOR THE VARIABLES MAXORD,MAXDEG,MAXVAL,MAXMU,MAXCOD,
#   AND CALLS PROCEDURE PKINIT TO PERFORM THE ACTUAL
#   INITIALIZATION.
#  PACKAGE PROCEDURE REQUIRED: PKINIT.
#  DECLARATION IN CALLING PROCEDURE: NONE.

   DEFLT = (5,20,5,1,4)
   PKINIT(DEFLT)

END  #  END OF PROCEDURE PINTT1.
```

```
PROCEDURE PINTT2
   INTEGER ARRAY DEFLT
#  THIS PROCEDURE SETS THE DEFAULT VALUES FOR SOLVER MODULE
#   SODET2 FOR THE VARIABLES MAXORD,MAXDEG,MAXVAL,MAXMU,MAXCOD,
#   AND CALLS PROCEDURE PKINIT TO PERFORM THE ACTUAL
#   INITIALIZATION.
#  PACKAGE PROCEDURE REQUIRED: PKINIT.
#  DECLARATION IN CALLING PROCEDURE: NONE.

   DEFLT = (5,20,5,1,4)
   PKINIT(DEFLT)

END  #  END OF PROCEDURE PINTT2.
```

```
PROCEDURE PINTNI
   INTEGER ARRAY DEFLT
#  THIS PROCEDURE SETS THE DEFAULT VALUES FOR SOLVER MODULE
#   SODENI FOR THE VARIABLES MAXORD,MAXDEG,MAXVAL,MAXMU,MAXCOD,
#   AND CALLS PROCEDURE PKINIT TO PERFORM THE ACTUAL
#   INITIALIZATION.
#  PACKAGE PROCEDURE REQUIRED: PKINIT.
#  DECLARATION IN CALLING PROCEDURE: NONE.

   DEFLT = (5,15,5,4,4)
   PKINIT(DEFLT)

END  #  END OF PROCEDURE PINTNI.
```

```
PROCEDURE SODEC1

#   MAIN SOLVER MODULE FOR THE CHEBYSHEV SERIES SOLUTION OF
#   A LINEAR ODE.
#   INPUT.
#       THE FOLLOWING INDETERMINATES MAY BE PRESENT IN THE INPUT:
#     X    - THE VARIABLE IN THE POLYNOMIAL COEFFICIENTS;
#     Y    - THE UNKNOWN SOLUTION FUNCTION;
#    DY    - DY(I) REPRESENTS THE I-TH DERIVATIVE OF Y;
#    YX    - YX(J) REPRESENTS THE VALUE OF Y AT A POINT X(J);
#   DYX    - DYX(I,J) REPRESENTS THE VALUE OF DY(I) AT A POINT X(J);
#    MU    - MU(J) REPRESENTS THE UNSPECIFIED INDETERMINATE APPEARING
#            IN THE PROBLEM;
#    THE INPUT DECK BEGINS WITH
#    NUM   - THE NUMBER OF PROBLEMS TO BE SOLVED.
#    FOR EACH PROBLEM THE INPUT SEQUENCE IS AS FOLLOWS:
#   ORDER  - THE ORDER OF THE ODE.
#   NMAX   - THE DEGREE OF POLYNOMIAL APPROXIMATION DESIRED;
#   RANGE  - THE FINITE INTERVAL (A,B) IN WHICH THE SOLUTION IS DESIRED;
#   CODE   - THE SOLUTION CODE FOR THE PROBLEM;
#   DIFFEQ- THE ODE AS A MULTINOMIAL IN THE INDETERMINATES X,Y,
#           DY(1),...,DY(ORDER);
#   CONDN - THE ASSOCIATED CONDITIONS CONDN(1),...,CONDN(ORDER-1) WRITTEN
#           AS MULTINOMIALS IN THE INDETERMINTATES YX(J),DYX(I,J),
#           1 <= J <= MAXSUB, 1 <= I <= ORDER-1, WHERE MAXSUB IS
#           THE NUMBER OF X-VALUES APPEARING IN CONDN;
#   XSUB   - RATIONAL NUMBERS XSUB(1),...,XSUB(MAXSUB) REPRESENTING
#            THE X-VALUES ABOVE.
#   OUTPUT:
#      IN ADDITION TO THE INDETERMINATES ABOVE, THE FOLLOWING
#      INDETERMINATES MAY APPEAR IN THE OUTPUT FROM THE PACKAGE:
#    R     - THE INDETERMINATE APPEARING AS THE INDEPENDENT VARIABLE IN
#            THE RECURRENCE EQUATION;
#    CR    - CR(J) REPRESENTS THE TERM C(R+J) IN THE RECURRENCE EQUATION
#            WHERE R REPRESENTS AN INDETERMINATE;
#    T     - T(J) REPRESENTS THE J-TH CHEBYSHEV POLYNOMIAL.
#   THE FIRST PART OF THE OUPUT IS AN ECHO OF THE INPUT.  THE REMAINDER
#   TO THE APPROXIMATE SOLUTION,ITS CHEBYSHEV COEFFICIENTS, AND THE
#   PERTURBATION IMPOSED ON THE ODE.  THE PACKAGE MAY PRINT OTHER
#   INFORMATION DEPENDING ON THE VALUE OF THE MESSAGE LEVEL
#   INDICATOR, MSGLVL, AS FOLLOWS:
#   MSGLVL
#     1     SUMMARY OF RESULTS OF COMPUTATION
#     2     MORE DETAILED INFORMATION
#     3     DETAILED INFORMATION PLUS TIMING STATISTICS
#     4     DETAILED DEBUGGING INFORMATION
#   PACKAGE PROCEDURES REQUIRED: FOWERS,PROC,SOLCLEPIC
#   SYSTEM PROCEDURE REQUIRED:  TIME
#   DECLARATION IN CALLING PROCEDURE:  NONE

  INTEGER ORDER,I,J,MAXSUB,NMAX,CODE,NUM
  INTEGER ARRAY XSUB(1:MAXSUB),RANGE(1:2)
  RATIONAL ARRAY XSUB,RANGE
  LOGICAL LINEAR
```

```
   SET UP GLOBAL LAYOUT AS LONG ALGEBRAIC GLOBL.

   LONG ALGEBRAIC (DY(0:MAXORD):YDEG,Y:YDEG,X:XDEG,
        MU(1:IMAX(1,MAXMU)):XDEG,DYX(1:MAXORD,1:MAXVAL):1,
        YX(1:MAXVAL):1,T(0:XDEG):1,PAR(0:MAXPAR):1,
        R:XDEG,TR(-MAXDEG:MAXDEG):1,CR(-MAXDEG:MAXDEG):1) GLOBL
   EXTERNAL LONG ALGEBRAIC XX=X, XY=Y , RR=R
   EXTERNAL LONG ALGEBRAIC ARRAY XT=T,XPAR=PAR
   EXTERNAL LONG ALGEBRAIC ARRAY XYX=YX,XCR=CR,XTR=TR
   EXTERNAL LONG ALGEBRAIC ARRAY XDYX=DYX,XMU=MU
   EXTERNAL LONG ALGEBRAIC ARRAY XDY=DY,XPOWER,TPOWER
   LONG ALGEBRAIC DIFFEQ=GLOBL,YNEW,RHS
   LONG ALGEBRAIC ARRAY CONDN=DIFFEQ,POLY,C
   REAL TOLD,T1,TNEW
   LONG ALGEBRAIC ALTRAN SOLC1
   LONG ALGEBRAIC ARRAY ALTRAN POWERS

   XPOWER = POWERS(XDEG)
   READ NUM
   DO I = 1,NUM
      PAGE()
      TIME(TOLD)
      T1 = TOLD
      PROC(NMAX,ORDER,DIFFEQ,CODE,POLY,RHS,MAXSUB,
           XSUB,CONDN,RANGE)
      TNEW = TIME(TOLD)
      IF(MSGLVL>2) WRITE " TIME USED FOR PROLOGUE WAS", TNEW

#  OBTAIN POLYNOMIAL SOLUTION, YNEW.

      YNEW = SOLC1(ORDER,POLY,NMAX,RHS,MAXSUB,CONDN,XSUB,CODE,C)

      TIME(TOLD)
      EPIC(NMAX,YNEW,C,RANGE)
      TNEW = TIME(TOLD)
      IF(MSGLVL>2) WRITE " TIME USED FOR EPILOGUE WAS", TNEW

      SYSERR:     #  CHECK ERROR STATE.
      IERR = HERRNO()
      IF (IERR<>0) DO
         IF(INPFLG<>1) DUMP()
         ELSE DO; SNAP; OK(); IERR=0; DOEND
       DOEND
      TNEW = TIME(T1)
      IF(MSGLVL>2) WRITE "TOTAL TIME ELAPSED WAS", TNEW
      WRITE " ", " ++++++++++++++++++++++++++++++++++++++++++++++++++", " "
   DOEND
   RETURN
END  #  END OF PROCEDURE SODEC1.
```

```
  SET UP GLOBAL LAYOUT AS LONG ALGEBRAIC GLOBL.

  LONG ALGEBRAIC (DY(0:MAXORD):YDEG,Y:YDEG,X:XDEG,
       MU(1:IMAX(1,MAXMU)):XDEG,DYX(1:MAXORD,1:MAXYAL):1,
       YX(1:MAXVAL):1,T(0:XDEG):1,PAR(0:MAXPAR):1,
       R:XDEG,TR(-MAXDEG:MAXDEG):1,CR(-MAXDEG:MAXDEG):1) GLOBL
  EXTERNAL LONG ALGEBRAIC XX=X, XY=Y , RR=R
  EXTERNAL LONG ALGEBRAIC ARRAY XT=T,XPAR=PAR
  EXTERNAL LONG ALGEBRAIC ARRAY XYX=YX,XCR=CR,XTR=TR
  EXTERNAL LONG ALGEBRAIC ARRAY XDYX=DYX,XMU=MU
  EXTERNAL LONG ALGEBRAIC ARRAY XDY=DY,XPOWER,TPOWER
  LONG ALGEBRAIC DIFFEQ=GLOBL,YNEW,RHS
  LONG ALGEBRAIC ARRAY CONDN=DIFFEQ,POLY,C
  REAL TOLD,T1,TNEW
  LONG ALGEBRAIC ALTRAN SOLC1
  LONG ALGEBRAIC ARRAY ALTRAN POWERS

  XPOWER = POWERS(XDEG)
  READ NUM
  DO I = 1,NUM
     PAGE()
     TIME(TOLD)
     T1 = TOLD
     PROC(NMAX,ORDER,DIFFEQ,CODE,POLY,RHS,MAXSUB,
          XSUB,CONDN,RANGE)
     TNEW = TIME(TOLD)
     IF(MSGLVL>2) WRITE " TIME USED FOR PROLOGUE WAS", TNEW

#  OBTAIN POLYNOMIAL SOLUTION, YNEW.

     YNEW = SOLC1(ORDER,POLY,NMAX,RHS,MAXSUB,CONDN,XSUB,CODE,C)

     TIME(TOLD)
     EPIC(NMAX,YNEW,C,RANGE)
     TNEW = TIME(TOLD)
     IF(MSGLVL>2) WRITE " TIME USED FOR EPILOGUE WAS", TNEW

     SYSERR:     #  CHECK ERROR STATE.
     IERR = HERRNO()
     IF (IERR<>0) DO
        IF(INPFLG<>1) DUMP()
        ELSE DO; SNAP; OK(); IERR=0; DOEND
      DOEND
     TNEW = TIME(T1)
     IF(MSGLVL>2) WRITE "TOTAL TIME ELAPSED WAS", TNEW
     WRITE " ", " ++++++++++++++++++++++++++++++++++++++++++++++++++", " "
  DOEND
  RETURN
END  #  END OF PROCEDURE SODEC1.
```

```
PROCEDURE SODET1

#   MAIN SOLVER MODULE FOR THE TAU METHOD SOLUTION OF A LINEAR
#   ODE, BY DIRECT SERIES SUBSTITUITION.
#   INPUT:
#      THE FOLLOWING INDETERMINATES MAY BE PRESENT IN THE INPUT:
#     X    - THE VARIABLE IN THE POLYNOMIAL COEFFICIENTS;
#     Y    - THE UNKNOWN SOLUTION FUNCTION;
#     DY   - DY(I) REPRESENTS THE I-TH DERIVATIVE OF Y;
#     YX   - YX(J) REPRESENTS THE VALUE OF Y AT A POINT X(J);
#     DYX  - DYX(I,J) REPRESENTS THE VALUE OF DY(I) AT A POINT X(J);
#     MU   -- MU(J) REPRESENTS THE UNSPECIFIED INDETERMINATE APPEARING
#            IN THE PROBLEM;
#     THE INPUT DECK BEGINS WITH
#     NUM  -- THE NUMBER OF PROBLEMS TO BE SOLVED.
#     FOR EACH PROBLEM THE INPUT SEQUENCE IS AS FOLLOWS:
#   ORDER -- THE ORDER OF THE ODE;
#   NMAX  -- THE DEGREE OF POLYNIOMIAL APPROXIMATION DESIRED;
#   RANGE - THE FINITE INTERVAL (A,B) IN WHICH THE SOLUTION IS DESIRED;
#   CODE  - THE SOLUTION CODE FOR THE PROBLEM;
#   DIFFEQ- THE ODE AS A MULTINOMIAL IN THE INDETERMINATES X,Y,
#           DY(1),.....,DY(ORDER);
#   CONDN - THE ASSOCIATED CONDITIONS CONDN(1),...,CONDN(ORDER) WRITTEN
#           AS MULTINOMIALS IN THE INDETERMINTATES YX(J),DYX(I,J),
#           1 <= J <= MAXSUB, 1 <= I <= ORDER-1, WHERE MAXSUB IS
#           THE NUMBER OF X--VALUES APPEARING IN CONDN;
#   XSUB  - RATIONAL NUMBERS XSUB(1),.....XSUB(MAXSUB) REPRESENTING
#           THE X--VALUES ABOVE.
#   OUTPUT:
#      IN ADDITION TO THE INDETERMINATES ABOVE, THE FOLLOWING
#        INDETERMINATES MAY APPEAR IN THE OUTPUT FROM THE PACKAGE:
#    R     -  THE INDETERMINATE APPEARING AS THE INDEPENDENT VARIABLE IN
#           THE RECURRENCE EQUATION;
#   AR     - AR(J) REPRESENTS THE TERM A(R+J) IN THE RECURRENCE EQUATION,
#           WHERE R REPRESENTS AN INDETERMINATE;
#   T      - T(J) REPRESENTS THE J-TH CHEBYSHEV POLYNOMIAL.
#   THE FIRST PART OF THE OUPUT IS AN ECHO OF THE INPUT.  IN ADDITION
#   TO THE APPROXIMATE SOLUTION,ITS CHEBYSHEV COEFFICIENTS, AND THE
#   PERTURBATION IMPOSED ON THE ODE, THE PACKAGE MAY PRINT OTHER
#   INFORMATION DEPENDING ON THE VALUE OF THE MESSAGE LEVEL
#   INDICATOR, MSGLVL, AS FOLLOWS:
#   MSGLVL
#     1      SUMMARY OF RESULTS OF COMPUTATION
#     2      MORE DETAILED INFORMATION
#     3      DETAILED INFORMATION PLUS TIMING STATISTICS
#     4      DETAILED DEBUGGING INFORMATION.
#   PACKAGE PROCEDURES REQUIRED: POWERS,TPOWS,PROT,SOLT1,EPIT
#   SYSTEM PROCEDURE REQUIRED:   TIME
#   DECLARATION IN CALLING PROCEDURE:  NONE.

   INTEGER ORDER,I,J,K,MAXSUB,NMAX,CODE,NUM
   EXTERNAL INTEGER MAXORD,MAXDEG,MAXVAL,MAXMU,XDEG,YDEG
   EXTERNAL INTEGER MAXPAR,MSGLVL,IERR,INPFLG
   RATIONAL ARRAY XSUB,RANGE
   LOGICAL LINEAR
```

```
#   SET UP GLOBAL LAYOUT AS LONG ALGEBRAIC GLOBL

    LONG ALGEBRAIC (DY(0:MAXORD):YDEG,Y:YDEG,X:XDEG,
         MU(1:IMAX(1,MAXMU)):XDEG,DYX(1:MAXORD,1:MAXVAL):1,
         YX(1:MAXVAL):1,T(0:XDEG):1,AR(-MAXDEG:MAXDEG):1,
         R:XDEG,PAR(0:MAXPAR):1,XR(-MAXDEG:MAXDEG):1) GLOBL
    EXTERNAL LONG ALGEBRAIC XX=X, XY=Y, RR=R
    EXTERNAL LONG ALGEBRAIC ARRAY XT=T,XAR=AR,XPAR=PAR
    EXTERNAL LONG ALGEBRAIC ARRAY XXR=XR,XYX=YX
    EXTERNAL LONG ALGEBRAIC ARRAY XDYX=DYX,XMU=MU
    EXTERNAL LONG ALGEBRAIC ARRAY XDY=DY,XPOWER,TPOWER
    LONG ALGEBRAIC DIFFEQ=GLOBL,YNEW,PERT,RHS
    LONG ALGEBRAIC ARRAY CONDN=DIFFEQ,POLY
    REAL TOLD,T1,TNEW
    LONG ALGEBRAIC ALTRAN SOLT1
    LONG ALGEBRAIC ARRAY ALTRAN POWERS,TPOWS

    XPOWER = POWERS(XDEG)
    TPOWER = TPOWS(XDEG,XX)

    READ NUM
    DO I = 1,NUM
       PAGE()
       TIME(TOLD)
       T1 = TOLD
       PROT(NMAX,ORDER,DIFFEQ,CODE,POLY,RHS,MAXSUB,
            XSUB,CONDN,RANGE)
       TNEW = TIME(TOLD)
       IF(MSGLVL>2) WRITE " TIME USED FOR PROLOGUE WAS", TNEW

    #  OBTAIN POLYNOMIAL SOLUTION, YNEW.

       YNEW = SOLT1(ORDER,POLY,NMAX,RHS,CODE,PERT,MAXSUB,CONDN,XSUB)

       TIME(TOLD)
       EPIT(NMAX,YNEW,PERT,RANGE)
       TNEW = TIME(TOLD)
       IF(MSGLVL>2) WRITE " TIME USED FOR EPILOGUE WAS", TNEW

       SYSERR:     #  CHECK ERROR STATE.
            IERR = HERRNO()
            IF (IERR<>0) DO
               IF(INPFLG<>1) DUMP()
               ELSE DO; SNAP; OK(); IERR=0; DOEND
            DOEND
       TNEW = TIME(T1)
       IF(MSGLVL>2) WRITE "TOTAL TIME ELAPSED WAS", TNEW
       WRITE " ", " ++++++++++++++++++++++++++++++++++++++++", "#"
    DOEND
    RETURN
END  #  END OF PROCEDURE SODET1.
```

```
#  MAIN SOLVER MODULE FOR THE TAU METHOD SOLUTION OF A LINEAR
#  ODE,VIA CANONICAL POLYNOMIALS.
#  INPUT:
#     THE FOLLOWING INDETERMINATES MAY BE PRESENT IN THE INPUT:
#    X    - THE VARIABLE IN THE POLYNOMIAL COEFFICIENTS;
#    Y    - THE UNKNOWN SOLUTION FUNCTION;
#   DY    - DY(I) REPRESENTS THE I'TH DERIVATIVE OF Y;
#   YX    - YX(J) REPRESENTS THE VALUE OF Y AT A POINT X(J);
#   DYX   - DYX(I,J) REPRESENTS THE VALUE OF DY(I) AT A POINT X(J);
#   MU    - MU(J) REPRESENTS THE UNSPECIFIED INDETERMINATE APPEARING
#           IN THE PROBLEM;
#   THE INPUT DECK BEGINS WITH
#   NUM   - THE NUMBER OF PROBLEMS TO BE SOLVED.
#   FOR EACH PROBLEM THE INPUT SEQUENCE IS AS FOLLOWS:
#  ORDER - THE ORDER OF THE ODE;
#  NMAX  - THE DEGREE OF POLYNOMIAL APPROXIMATION DESIRED;
#  RANGE - THE FINITE INTERVAL (A,B) IN WHICH THE SOLUTION IS DESIRED;
#  CODE  - THE SOLUTION CODE FOR THE PROBLEM;
#  DIFFEQ- THE ODE AS A MULTINOMIAL IN THE INDETERMINATES X,Y,
#          DY(1),.....,DY(ORDER);
#  CONDN - THE ASSOCIATED CONDITIONS CONDN(1),...,CONDN(ORDER) WRITTEN
#          AS MULTINOMIALS IN THE INDETERMINATES YX(J),DYX(I,J),
#          I <= J <= MAXSUB, 1 <= I <= ORDER-1, WHERE MAXSUB IS
#          THE NUMBER OF X-VALUES APPEARING IN CONDN;
#  XSUB  - RATIONAL NUMBERS XSUB(1),.....XSUB(MAXSUB) REPRESENTING
#          THE X-VALUES ABOVE.
#  OUTPUT:
#     IN ADDITION TO THE INDETERMINATES ABOVE, THE FOLLOWING
#       INDETERMINATES MAY APPEAR IN THE OUTPUT FROM THE PACKAGE:
#   R     -  THE INDETERMINATE APPEARING AS THE INDEPENDENT VARIABLE IN
#            THE RECURRENCE EQUATION;
#  AR     -  AR(J) REPRESENTS THE TERM A(R+J) IN THE RECURRENCE EQUATION,
#            WHERE R REPRESENTS AN INDETERMINATE;
#  T      -  T(J) REPRESENTS THE J-TH CHEBYSHEV POLYNOMIAL.
#  THE FIRST PART OF THE OUPUT IS AN ECHO OF THE INPUT.  IN ADDITION
#  TO THE APPROXIMATE SOLUTION,ITS CHEBYSHEV COEFFICIENTS, AND THE
#  PERTURBATION IMPOSED ON THE ODE, THE PACKAGE MAY PRINT OTHER
#  INFORMATION DEPENDING ON THE VALUE OF THE MESSAGE LEVEL
#  INDICATOR, MSGLVL, AS FOLLOWS:
#  MSGLVL
#    1      SUMMARY OF RESULTS OF COMPUTATION
#    2      MORE DETAILED INFORMATION
#    3      DETAILED INFORMATION PLUS TIMING STATISTICS
#    4      DETAILED DEBUGGING INFORMATION.
#  PACKAGE PROCEDURES REQUIRED: POWERS,TPOWS,PROJ,SOLT2,EPIT,COM
#  SYSTEM PROCEDURE REQUIRED:   TIME
#  DECLARATION IN CALLING PROCEDURE:  NONE.


  INTEGER ORDER,I,J,K,MAXSUB,NMAX,CODE,NUM
  EXTERNAL INTEGER IERR,MAXORD,MAXDEG,MAXVAL,MAXMU,XDEG,YDEG
  EXTERNAL INTEGER MAXPAR,MSGLVL,IERR,INPFLG
  INTEGER MM=MAXDEG+MAXPAR
  RATIONAL ARRAY XSUB,RANGE
  LOGICAL LINEAR
```

SET UP GLOBAL LAYOUT AS LONG ALGEBRAIC GLOBL.


```
LONG ALGEBRAIC (DY(0:MAXORD):YDEG,Y:YDEG,X:XDEG,
     MU(1:IMAX(1,MAXMU)):XDEG,DYX(1:MAXORD,1:MAXVAL):1,
     YX(1:MAXVAL):1,T(0:XDEG):1,QR(-MM:0):1,UQ(0:MM):1,
     R:XDEG,PAR(0:MAXPAR):1,XR(-MAXDEG:MAXDEG):1) GLOBL
EXTERNAL LONG ALGEBRAIC XX=X, XY=Y, RR=R
EXTERNAL LONG ALGEBRAIC ARRAY XT=T,XQR=QR,XPAR=PAR
EXTERNAL LONG ALGEBRAIC ARRAY XXR=XR,XUQ=UQ,XYX=YX
EXTERNAL LONG ALGEBRAIC ARRAY XDYX=DYX,XMU=MU
EXTERNAL LONG ALGEBRAIC ARRAY XDY=DY,XPOWER,TPOWER
LONG ALGEBRAIC DIFFEQ=GLOBL,YNEW,PERT,RHS
LONG ALGEBRAIC ARRAY CONDN=DIFFEQ,POLY
REAL TOLD,T1,TNEW
LONG ALGEBRAIC ALTRAN SOLT2
LONG ALGEBRAIC ARRAY ALTRAN POWERS,TPOWS

XPOWER = POWERS(XDEG)
TPOWER = TPOWS(XDEG,XX)

READ NUM
DO I = 1,NUM
   PAGE()
   TIME(TOLD)
   T1 = TOLD
   PROT(NMAX,ORDER,DIFFEQ,CODE,POLY,RHS,MAXSUB,
        XSUB,CONDN,RANGE)
   TNEW = TIME(TOLD)
   IF(MSGLVL>2) WRITE " TIME USED FOR PROLOGUE WAS", TNEW

#  OBTAIN POLYNOMIAL SOLUTION, YNEW.

   YNEW = SOLT2(ORDER,POLY,NMAX,RHS,CODE,PERT,MAXSUB,CONDN,XSUB)

   TIME(TOLD)
   EPIT(NMAX,YNEW,PERT,RANGE)
   TNEW = TIME(TOLD)
   IF(MSGLVL>2) WRITE " TIME USED FOR EPILOGUE WAS", TNEW

   SYSERR:     #  CHECK ERROR STATE.
   IERR = HERRNO()
   IF (IERR<>0) DO
      IF(INPFLG<>1) DUMP()
      ELSE DO; SNAP; OK(); IERR=0; DOEND
   DOEND
   TNEW = TIME(T1)
   IF(MSGLVL>2) WRITE "TOTAL TIME ELAPSED WAS", TNEW
   WRITE " ", " ++++++++++++++++++++++++++++++++++++++++++", " "
 DOEND
 RETURN
END  #  END OF PROCEDURE SODET2.
```

```
#  MAIN SOLVER MODULE FOR THE TAU METHOD SOLUTION OF A LINEAR
#  OR NONLINEAR ODE, BY DIRECT SERIES SUBSTITUTION
#  A FORM OF NEWTON ITERATION IS USED TO REDUCE THE
#  TO THAT OF SOLVING SUCCESSIVE LINEAR ODES.
#  INPUT:
#     THE FOLLOWING INDETERMINATES MAY BE PRESENT IN THE INPUT:
#   X    - THE VARIABLE IN THE POLYNOMIAL COEFFICIENTS;
#   Y    - THE UNKNOWN SOLUTION FUNCTION;
#  DY    - DY(I) REPRESENTS THE I-TH DERIVATIVE OF Y;
#  YX    - YX(J) REPRESENTS THE VALUE OF Y AT A POINT X(J);
# DYX    - DYX(I,J) REPRESENTS THE VALUE OF DY(I) AT A POINT X(J);
#  MU    - MU(J) REPRESENTS THE UNSPECIFIED INDETERMINATE APPEARING
#          IN THE PROBLEM;
#  THE INPUT DECK BEGINS WITH
#  NUM   - THE NUMBER OF PROBLEMS TO BE SOLVED.
#  FOR EACH PROBLEM THE INPUT SEQUENCE IS AS FOLLOWS:
# ORDER  - THE ORDER OF THE ODE;
# NMAX   - THE DEGREE OF POLYNOMIAL APPROXIMATION DESIRED;
# RANGE  - THE FINITE INTERVAL (A,B) IN WHICH THE SOLUTION IS DESIRED;
# CODE   - THE SOLUTION CODE FOR THE PROBLEM;
# DIFFEQ - THE ODE AS A MULTINOMIAL IN THE INDETERMINATES X,Y,
#          DY(1),.....,DY(ORDER);
# CONDN  - THE ASSOCIATED CONDITIONS CONDN(1),...,CONDN(ORDER) WRITTEN
#          AS MULTINOMIALS IN THE INDETERMINTATES YX(J),DYX(I,J),
#          1 <= J <= MAXSUB, 1 <= I <= ORDER-1, WHERE MAXSUB IS
#          THE NUMBER OF X-VALUES APPEARING IN CONDN;
# XSUB   - RATIONAL NUMBERS XSUB(1), ...XSUB(MAXSUB) REPRESENTING
#          THE X-VALUES ABOVE.
# OUTPUT:
#    IN ADDITION TO THE INDETERMINATES ABOVE, THE FOLLOWING
#      INDETERMINATES MAY APPEAR IN THE OUTPUT FROM THE PACKAGE:
#  R     -  THE INDETERMINATE APPEARING AS THE INDEPENDENT VARIABLE IN
#           THE SUCCESSIVE RECURRENCE EQUATIONS;
# AR     - AR(J) REPRESENTS THE TERM A(R+J) IN THE RECURRENCE EQUATION,
#          WHERE R REPRESENTS AN INDETERMINATE;
# T      - T(J) REPRESENTS THE J-TH CHEBYSHEV POLYNOMIAL.
# THE FIRST PART OF THE OUPUT IS AN ECHO OF THE INPUT.  IN ADDITION
# TO THE APPROXIMATE SOLUTION,ITS CHEBYSHEV COEFFICIENTS, AND THE
# PERTURBATION IMPOSED ON THE ODE, THE PACKAGE MAY PRINT OTHER
# INFORMATION DEPENDING ON THE VALUE OF THE MESSAGE LEVEL
# INDICATOR, MSGLVL, AS FOLLOWS:
# MSGLVL
#   1      SUMMARY OF RESULTS OF COMPUTATION
#   2      MORE DETAILED INFORMATION
#   3      DETAILED INFORMATION PLUS TIMING STATISTICS
#   4      DETAILED DEBUGGING INFORMATION.
#  WHEN MSGLVL>1, THE RELEVANT INFORMATION IS PRINTED
#  AT EACH ITERATION.
# PACKAGE PROCEDURES REQUIRED: POWERS,TPOWS,PRON,SOLN1,EPIN
# SYSTEM PROCEDURE REQUIRED:   TIME
# DECLARATION IN CALLING PROCEDURE:  NONE

  INTEGER ORDER,I,J,K,MAXSUB,NMAX,CODE,NUM,NITER,CURN,NUMIT
  EXTERNAL INTEGER MAXORD,MAXDEG,MAXVAL,MAXMU,XDEG,YDEG
  EXTERNAL INTEGER MAXPAR,MSGLVL,IERR,INPFLG
  RATIONAL ARRAY XSUB,RANGE
  LOGICAL LINEAR
```

```
 SET UP GLOBAL LAYOUT AS LONG ALGEBRAIC GLOBL.

LONG ALGEBRAIC (DY(0:MAXORD):YDEG,Y:YDEG,X:XDEG,
      MU(1:IMAX(1,MAXMU)):XDEG,DYX(1:MAXORD,1:MAXVAL):1,
      YX(1:MAXVAL):1,T(0:XDEG):1,AR(-MAXDEG:MAXDEG):1,
      R:XDEG,PAR(0:MAXPAR):1,XR(-MAXDEG:MAXDEG):1) GLOBL
EXTERNAL LONG ALGEBRAIC XX=X, XY=Y, RR=R
EXTERNAL LONG ALGEBRAIC ARRAY XT=T,XAR=AR,XPAR=PAR
EXTERNAL LONG ALGEBRAIC ARRAY XXR=XR,XYX=YX
EXTERNAL LONG ALGEBRAIC ARRAY XDYX=DYX,XMU=MU
EXTERNAL LONG ALGEBRAIC ARRAY XDY=DY,XPOWER,TPOWER
LONG ALGEBRAIC DIFFEQ=GLOBL,YNEW,PERT,RHS,YINIT
LONG ALGEBRAIC ARRAY CONDN=DIFFEQ,POLY
REAL TOLD,T1,TNEW
LONG ALGEBRAIC ALTRAN SOLN1
LONG ALGEBRAIC ARRAY ALTRAN POWERS,TPOWS

 XPOWER  = POWERS(XDEG)
 TPOWER  = TPOWS(XDEG,XX)

 READ NUM
 DO I = 1,NUM
    PAGE()
    TIME(TOLD)
    T1 = TOLD
    PRON(CURN,NMAX,NUMIT,ORDER,DIFFEQ,POLY,MAXSUB,
         XSUB,CONDN,CODE,RHS,YINIT,RANGE)
    TNEW = TIME(TOLD)
    IF(MSGLVL>2) WRITE " TIME USED FOR PROLOGUE WAS", TNEW

 #  OBTAIN POLYNOMIAL SOLUTION, YNEW.

    YNEW = SOLN1(ORDER,POLY,DIFFEQ,RHS,CURN,NMAX,NUMIT,
                 NITER,CODE,PERT,MAXSUB,CONDN,XSUB,YINIT)

    TIME(TOLD)
    EPIN(NMAX,NITER,YNEW,PERT,RANGE)
    TNEW = TIME(TOLD)
    IF(MSGLVL>2) WRITE " TIME USED FOR EPILOGUE WAS", TNEW

    SYSERR:      # CHECK ERROR STATE.
         IERR = HERRNO()
         IF (IERR<>0) DO
            IF(INPFLG<>1) DUMP()
            ELSE DO; SNAP; OK(); IERR=0; DOEND
         DOEND
     TNEW = TIME(T1)
    IF(MSGLVL>2) WRITE "TOTAL TIME ELAPSED WAS", TNEW
    WRITE " ", "++++++++++++++++++++++++++++++++++++++++++", " "
  DOEND
  RETURN
END  #  END OF PROCEDURE SODEN1.
```

List of Level 2 Procedures

List of Level 2 Procedures

*Not a Level 2 Procedure, but included here because it performs most of the work required in PROC and PROT.

```
PROCEDURE PROC(N,ORDER,DIFFEQ,CODE,POLY,
               RHS,MAXSUB,XSUB,CONDN,RANGE)
  INTEGER N,ORDER,MAXSUB,CODE
  LOGICAL LINEAR
  RATIONAL ARRAY XSUB
  RATIONAL ARRAY RANGE
  LONG ALGEBRAIC DIFFEQ,RHS
  LONG ALGEBRAIC ARRAY POLY,CONDN

#       PROLOGUE ROUTINE FOR CHEBYSHEV SERIES METHODS.
# THIS PROCEDURE CALLS PROCEDURE PROLGL TO READ IN THE INPUT
# FOR A LINEAR ODE AND ITS ASSOCIACED CONDITIONS.
# PACKAGE PROCEDURE REQUIRED: PROLGL.
# DECLARATION IN CALLING PROCEDURE: NONE.

  PROLGL(N,ORDER,DIFFEQ,CODE,POLY,RHS,
         MAXSUB,XSUB,CONDN,RANGE)

  RETURN
END # END OF PROCEDURE PROC.
```

```
      PROCEDURE PROT(N,ORDER,DIFFEQ,CODE,POLY,
     *               RHS,MAXSUB,XSUB,CONDN,RANGE)
        INTEGER N,ORDER,MAXSUB,CODE
        LOGICAL LINEAR
        RATIONAL ARRAY XSUB
        RATIONAL ARRAY RANGE
        LONG ALGEBRAIC DIFFEQ,RHS
        LONG ALGEBRAIC ARRAY POLY,CONDN

C               PROLOGUE ROUTINE FOR TAU METHODS.
C     THIS PROCEDURE CALLS PROCEDURE PROLGL TO READ IN THE INPUT
C     FOR A LINEAR ODE AND ITS ASSOCIACED CONDITIONS.
C     PACKAGE PROCEDURE REQUIRED: PROLGL
C     DECLARATION IN CALLING PROCEDURE: NONE.

        PROLGL(N,ORDER,DIFFEQ,CODE,POLY,RHS,
     *         MAXSUB,XSUB,CONDN,RANGE)

        RETURN
      END # END OF PROCEDURE PROT.
```

```
PROCEDURE PRON(N,NMAX,NUMIT,ORDER,DIFFEQ,POLY,MAXSUB,
              XSUB,CONDN,CODE,RHS,YINIT,RANGE)
   INTEGER N,NMAX,NUMIT,ORDER,MAXSUB,CODE
   RATIONAL ARRAY XSUB,RANGE
   LONG ALGEBRAIC DIFFEQ,RHS,YINIT
   LONG ALGEBRAIC ARRAY POLY,CONDN

#  THIS PROCEDURE READS IN AN ODE AND ITS ASSOCIATED CONDITIONS AND
#  CHECKS THE VALIDITY OF THE ORDER, THE DEGREE OF APPROXIMATION
#  REQUIRED, THE SOLUTION CODE, AND THE RANGE IN WHICH THE SOLUTION
#  IS REQUIRED.
#  THE ACTUAL INPUT AND PARAMETER CHECKING ARE PERFORMED BY
#  A CALL TO PROCEDURE READER.
#  INPUT PARAMETERS: NONE.
#  OUTPUT PARAMTERS:
#    NMAX   - DEGREE OF APPROXIMATION REQUIRED;
#    NUMIT  - NUMBER OF ITERATIONS REQUIRED;
#    N      - DEGREE OF APPROXIMATION IN FIRST ITERATION.
#    ORDER  - ORDER OF ODE;
#    MAXSUB - NUMBER OF X-VALUES IN ASSOCIATED CONDITIONS;
#    CODE   - SOLUTION CODE;
#    XSUB   - AN ARRAY XSUB(1),...,XSUB(MAXSUB) OF X-VALUES APPEARING
#             IN THE ASSOCIATED CONDITIONS;
#    RANGE  - A RATIONAL ARRAY (A,B) INDICATING THE FINITE INTERVAL
#             IN WHICH THE SOLUTION IS DESIRED;
#    DIFFEQ - THE ODE EXPRESSED AS DIFFEQ=0;
#    POLY   - ARRAY POLY(0),...,POLY(ORDER) OF POLYNOMIAL COEFFICIENTS
#             IN ODE;
#    RHS    - RIGHT-HAND-SIDE POLYNOMIAL OF ODE;
#    CONDN  - ARRAY CONDN(1),...,CONDN(ORDER) OF ASSOCIATED CONDITIONS;
#    YINIT  - INITIAL APPROXIMATION SATISFYING ASSOCIATED CONDITIONS.
#  EXTERNAL VARIABLES REQUIRED: MSGLVL,XX.
#  PACKAGE PROCEDURES REQUIRED: READER, SETP1,STEP2,INIT.
#  DECLARATION IN CALLING PROCEDURE: NONE.

   INTEGER K
   LOGICAL LINEAR
   EXTERNAL INTEGER MSGLVL
   EXTERNAL LONG ALGEBRAIC XX
   LONG ALGEBRAIC ALTRAN INIT
   LONG ALGEBRAIC ARRAY ALTRAN SETP1,SETP2
```

```
# READ IN AND CHECK ODE AND ASSOCIATED CONDITIONS.
#
   READER(NMAX,ORDER,DIFFEQ,CODE,MAXSUB,XSUB,CONDN,RANGE,LINEAR)

# DISTINGUISH BETWEEN LINEAR AND NONLINEAR ODE.

   IF (LINEAR) DO       # LINEAR ODE.
      NUMIT = 1
      WRITE " ODE IS LINEAR: ONLY 1 ITERATION NEEDED"
      CODE = 1
      WRITE " CODE SET TO 1"
      N = NMAX
      YINIT = 0
      POLY = SETUP(ORDER,DIFFEQ,RHS)
   ENDO
   ELSE DO              # NONLINEAR ODE
      NUMIT = 1; K = 1
      AGAIN: NUMIT = NUMIT + 1; K = K*2
             IF (K<NMAX) GO TO AGAIN
      N = 1
      YINIT = INITGUESS(ORDER,NMAX,MAXSUB,XSUB,CONDN,X0)
      POLY = SETUP(NORDER,N,DIFFEQ,YINIT,RHS)
   ENDO

   IF (MSGLVL >= 0) DO
      WRITE " THE INITIAL APPROXIMATION IS", YINIT
      WRITE " THE POLYNOMIAL COEFFICIENTS OF THE ODE ARE:",POLY
      WRITE "THE RIGHT HAND SIDE POLYNOMIAL IS:", RHS
   ENDO

   RETURN
END # END OF PROCEDURE PRON.
```

```
PROCEDURE SOLC1(ORDER,POLY,N,RHS,MAXSUB,CONDN,XSUB,CODE,COEF)
      INTEGER VALUE N,ORDER,MAXSUB,CODE
      RATIONAL ARRAY VALUE XSUB
      LONG ALGEBRAIC VALUE RHS
      LONG ALGEBRAIC ARRAY COEF
      LONG ALGEBRAIC ARRAY VALUE POLY,CONDN

    #  THIS PROCEDURE COMPUTES A (FINITE) CHEBYSHEV SERIES SOLUTION
    #  OF A LINEAR ODE, BY DIRECT SERIES SUBSTITUTION.
    #  INPUT PARAMTERS:
    #    N    - DEGREE OF APPROXIMATION DESIRED;
    #    ORDER - ORDER OF ODE;
    #    CODE  - THE METHOD OF SOLUTION DESIRED IS INDICATED
    #            BY THE VALUE OF CODE AS FOLLOWS:
    #            CODE = 1  - SOLUTION BY BACKWARD RECURRENCE
    #            CODE = 2  - SOLUTION BY GAUSSIAN ELIMINATION;
    #    DIFFEQ,POLY,RHS - INFORMATION ABOUT THE ODE;
    #    MAXSUB,XSUB,CONDN - INFORMATION ABOUT THE ASSOCIATED
    #                        CONDITIONS;
    #    OUTPUT PARAMETERS:
    #       COEF - THE REQUIRED CHEBYSHEV COEFFICIENTS;
    #       YSTAR - THE POLYNOMIAL APPROXIMATION (RETURNED).
    #  EXTERNAL VARIABLES REQUIRED: MSGLVL,XX.
    #  PACKAGE PROCEDURES REQUIRED: GSC11,GSC12,RECC11,CHFORM.
    #  SYSTEM PROCEDURES REQUIRED: TIME, DEG,PINTN.
    #  DECLARATION IN CALLING PROCEDURE: LONG ALGEBRAIC ALTRAN SOLC1.

      INTEGER HALFN,DEGRHS
      REAL TOLD,TNEW
      LONG ALGEBRAIC EQN,YSTAR
      EXTERNAL INTEGER MSGLVL
      EXTERNAL LONG ALGEBRAIC XX
      LONG ALGEBRAIC ALTRAN RECC11,CHFORM,PINTN
      LONG ALGEBRAIC ALTRAN GSC11,GSC12

#     OBTAIN RECURRENCE EQUATION.

      TIME(TOLD)
      EQN = RECC11(ORDER,POLY,HALFN)
      IF(MSGLVL>1) WRITE " THE RECURRENCE EQUATION IS" , EQN
      TNEW = TIME(TOLD)
      IF(MSGLVL>2)_
      WRITE "TIME TO GENERATE RECURRENCE EQUATION WAS", TNEW

#     OBTAIN THE CHEBYSHEV COEFFICIENTS AND POLYNOMIAL APPROXIMATION

      IF (CODE==1) DO
         YSTAR = GSC11(N,ORDER,EQN,HALFN,MAXSUB,CONDN,XSUB,RHS,COEF)
      DOEND
      ELSE IF (CODE==2) DO
         YSTAR = GSC12(N,ORDER,EQN,HALFN,MAXSUB,CONDN,XSUB,RHS,COEF)
      DOEND

      RETURN(YSTAR)
  END #  END OF PROCEDURE SOLC1.
```

```
PROCEDURE SOLT(ORDER,POLY,N,RHS,CODE,PERT,MAXSUB,CONDN,XSUB)
  INTEGER VALUE ORDER,N,MAXSUB,CODE
  RATIONAL ARRAY VALUE XSUB
  LONG ALGEBRAIC VALUE RHS
  LONG ALGEBRAIC ARRAY VALUE CONDN
  LONG ALGEBRAIC PERT
  LONG ALGEBRAIC ARRAY POLY

#  THIS PROCEDURE COMPUTES THE EXACT POLYNOMIAL SOLUTION OF
#  A PERTURBED LINEAR ODE, BY DIRECT SERIES SUBSTITUTION.
#  INPUT PARAMETERS:
#    N      - DEGREE OF APPROXIMATION DESIRED;
#    ORDER      - ORDER OF ODE;
#    POLY       - POLYNOMIAL COEFFICIENTS OF ODE;
#    RHS        - RIGHT-HAND-SIDE POLYNOMIAL OF ODE;
#    MAXSUB,XSUB,CONDN -- INFORMATION ABOUT ASSOCIATED CONDITIONS;
#    CODE       - THE FORM IN WHICH THE ODE IS TO BE SOLVED
#                 IS INDICATED BY THE VALUE OF CODE AS FOLLOWS:
#                 CODE = 1 INTEGRATED FORM WITH ORIGINAL CONDITIONS
#                 CODE = 2 INTEGRATED FORM WITH PERTURBED CONDITIONS
#                 CODE = 3 DIFFERENTIATED FORM WITH ORIGINAL CONDITIONS
#                 CODE = 4 DIFFERENTIATED FORM WITH PERTURBED CONDITIONS.
#
#  OUTPUT PARAMETERS:
#    YSTAR    - THE REQUIRED POLYNOMIAL APPROXIMATION(RETURNED);
#    PERT     - PERTURBATION ON ODE.
#  EXTERNAL VARIABLES REQUIRED: MSGLVL.
#  PACKAGE PROCEDURES REQUIRED: GST12, RECT12, RECT11.
#  SYSTEM PROCEDURES REQUIRED: TIME.
#  DECLARATION IN CALLING PROCEDURE: LONG ALGEBRAIC ALTRAN SOLT1.


    INTEGER LH1,H2
    REAL TOLD,TNEW
    LONG ALGEBRAIC YSTAR,EQN
    EXTERNAL INTEGER MSGLVL
    LONG ALGEBRAIC ALTRAN GST12,RECT12,RECT11

  #  OBTAIN RECURRENCE EQUATION.

    TIME(TOLD)
    EQN = RECT12(POLY,ORDER,H1,H2,CODE)
    TNEW = TIME(TOLD)
    IF(MSGLVL>1) WRITE " THE RECURRENCE EQUATION IS", EQN
    IF (MSGLVL>2)_
    WRITE "TIME TO GENERATE RECURRENCE EQUATION WAS", TNEW

  #  OBTAIN POLYNOMIAL SOLUTION AND PERTURBATION.

    YSTAR = GST12(EQN,N,H1,H2,CODE,PERT,MAXSUB,CONDN,XSUB,RHS,ORDER)

    RETURN(YSTAR)
END # END OF PROCEDURE SOLT1.
```

```
PROCEDURE SOLT2(ORDER,POLY,N,RHS,CODE,PERT,MAXSUB,CONDN,XSUB)
   INTEGER VALUE ORDER,N,MAXSUB,CODE
   RATIONAL ARRAY VALUE XSUB
   LONG ALGEBRAIC VALUE RHS
   LONG ALGEBRAIC ARRAY VALUE CONDN
   LONG ALGEBRAIC PERT
   LONG ALGEBRAIC ARRAY POLY

#  THIS PROCEDURE COMPUTES THE EXACT POLYNOMIAL SOLUTION OF
#  A PERURBED LINEAR ODE, USING THE CANONICAL POLYNOMIALS
#  ASSOCIATED WITH THE LINEAR OPERATOR DEFINED BY THE ODE.
#  INPUT PARAMETERS:
#    N      - DEGREE OF APPROXIMATION DESIRED;
#    ORDER      - ORDER OF ODE;
#    POLY       - POLYNOMIAL COEFFICIENTS OF ODE;
#    RHS        - RIGHT-HAND-SIDE POLYNOMIAL OF ODE;
#    MAXSUB,XSUB,CONDN - INFORMATION ABOUT ASSOCIATED CONDITIONS;
#    CODE    - THE FORM IN WHICH THE ODE IS TO BE SOLVED
#              IS INDICATED BY THE VALUE OF CODE AS FOLLOWS:
#              CODE = 1 INTEGRATED FORM WITH ORIGINAL CONDITIONS
#              CODE = 2 INTEGRATED FORM WITH PERTURBED CONDITIONS
#              CODE = 3 DIFFERENTIATED FORM WITH ORIGINAL CONDITIONS
#              CODE = 4 DIFFERENTIATED FORM WITH PERTURBED CONDITIONS.
#  OUTPUT PARAMETERS:
#     YSTAR   - THE REQUIRED POLYNOMIAL APPROXIMATION(RETURNED);
#     PERT    -- PERTURBATION ON ODE.
#  EXTERNAL VARIABLES REQUIRED: MSGLVL.
#  PACKAGE PROCEDURES REQUIRED: GST21,RECT21.
#  SYSTEM PROCEDURES REQUIRED: TIME.
#  DECLARATION IN CALLING PROCEDURE: LONG ALGEBRAIC ALTRAN SOLT2.

   INTEGER I,H1,H2
   REAL TOLD,TNEW
   LONG ALGEBRAIC YSTAR,EQN,GENPOL
   EXTERNAL INTEGER MSGLVL
   LONG ALGEBRAIC ALTRAN GST21,RECT21

#  OBTAIN THE RECURRENCE EQUATION DEFINING THE CANONICAL POLYNOMIALS.

   TIME(TOLD)
   EQN = RECT21(POLY,ORDER,H1,H2,CODE,GENPOL)
   IF (MSGLVL>1) WRITE " THE RECURRENCE EQUATION IS" , EQN
   TNEW = TIME(TOLD)
   IF (MSGLVL>2)_
   WRITE "TIME TO GENERATE RECURRENCE EQUATION WAS", TNEW

#  OBTAIN POLYNOMIAL SOLUTION AND PERTURBATION.

   YSTAR = GST21(N,ORDER,POLY,CODE,H1,H2,EQN,GENPOL,
                 RHS,MAXSUB,CONDN,XSUB,PERT)

   RETURN(YSTAR)
END #  END OF PROCEDURE SOLT2.
```

```
PROCEDURE SOLN1(ORDER,POLY,DIFFEQ,RHS,CURN,NMAX,NUMIT,
        NITER,CODE,PERT,MAXSUB,CONDN,XSUB,YINIT)
  INTEGER NITER,CURN
  INTEGER VALUE ORDER,NMAX,NUMIT,MAXSUB,CODE
  RATIONAL ARRAY VALUE XSUB
  LONG ALGEBRAIC VALUE DIFFEQ,RHS,YINIT
  LONG ALGEBRAIC ARRAY VALUE POLY,CONDN
  LONG ALGEBRAIC PERT

#  THIS PROCEDURE COMPUTES A POLYNOMIAL APPROXIMATION TO
#    A GIVEN LINEAR OR NONLINEAR ODE BY A FORM OF NEWTON
#    ITERATION.
#  INPUT PARAMETERS:
#    ORDER     -  ORDER OF THE ODE;
#    NMAX      -  DEGREE OF APPROXIMATION REQUIRED;
#    CURN      -  THE DEGREE OF APPROX. IN FIRST ITERATION;
#    NUMIT     -  NUMBER OF ITERATIONS REQUIRED;
#    CODE      -  SOLUTION CODE INDICATING MODIFICATIONS
#                 TO BE PERFORMED ON SUCCESSIVE ITERATES AS
#                 FOLLWS:
#                 CODE = 1  NO ROUNDING OR SHORTENING
#                 CODE = 2  ROUNDING ONLY
#                 CODE = 3  SHORTENING ONLY
#                 CODE = 4  ROUNDING AND SHORTENING;
#    DIFFEQ    -  THE ODE REPRESENTED AS DIFFEQ=0;
#    POLY -  POLYNOMIAL COEFFICIENTS OF INITIAL LINEAR ODE;
#    RHS       -  RIGHT-HAND-SIDE POLYNOMIAL OF INITIAL LINEAR ODE;
#    CONDN,XSUB,MAXSUB -  CONDITIONS ASSOCIATED WITH ODE;
#    YINIT     -  THE INITIAL APPROXIMATION TO THE SOLUTION FUNCTION.
#  OUTPUT PARAMETERS:
#    CURN      -  DEGREE OF LAST APPROXIMATION OBTAINED.  IF THE
#                 PROCEDURE TERMINATES NORMALLY, CURN=NMAX;
#    NITER     -  NUMBER OF ITERATIONS PERFORMED.  IF THE PROCEDURE
#                 TERMINATES NORMALLY, NITER=NUMIT;
#    YNEW      -  THE LAST POLYNOMIAL APPROXIMATION COMPUTED;
#    PERT      -  PERTURBATION.  IF THE PROCEDURE TERMINATES NORMALLY,
#                 PERT IS THE PERTURBATION ON THE ORIGINAL ODE.
#                 ELSE PERT IS THE PERTURBATION ON THE LAST LINEAR
#                 SOLVED.
#  EXTERNAL VARIABLES REQUIRED: MAXMU,MSGLVL,XMU.
#  PACKAGE PROCEDURES REQUIRED: SOLTI, NEXTIT.
#  SYSTEM PROCEDURES REQUIRED: ANUM, CTPO.
#  DECLARATION IN CALLING PROCEDURE: LONG ALGEBRAIC ALTRAN SOLN1.

  INTEGER I,CURIT,SC=1
  LONG ALGEBRAIC YNEW=YINIT,NUM,DEN,YCORR
  LONG INTEGER ALTRAN CTPO
  LONG ALGEBRAIC ALTRAN SOLTI
  EXTERNAL INTEGER MAXMU,MSGLVL
  EXTERNAL LONG ALGEBRAIC ARRAY XMU
```

```
#  OBTAIN CONDITIONS FOR CORRECTION TERM.

   WRITE "CONDITIONS FOR THE CORRECTION TERM ARE:"
   DO I = 1,ORDER
      CONDN(I) = CONDN(I)(XMU=1MAX(1,MAXMU)$(0))
      NUM = ANUM(CONDN(I),DEN)
      CONDN(I) = (NUM - CTPO(EXPAND(NUM)))/DEN
      WRITE CONDN(I)
   DOEND

   DO CURIT = 1,NUMIT
      NITER = CURIT
      IF (MSGLVL>1) DO
         PAGE()
         WRITE " ", " "
         WRITE "          ******* NEXT     ITERATION  *******"
         WRITE " ", " "
         WRITE "ITERATION NUMBER IS", CURIT
      DOEND

      #   OBTAIN CORRECTION TERM.

      YCORR = SOLT(ORDER,POLY,CURN,RHS,
                   SC,PERT,MAXSUB,CONDN,XSUB)

      #  PREPARE FOR THE NEXT ITERATION.

      NEXTIT(CURN,NMAX,ORDER,YNEW,YCORR,POLY,PERT,CODE,
             DIFFEQ,RHS)
   DOEND

   RETURN(YNEW)
END # END OF PROCEDURE SOLN1.
```

```
PROCEDURE EPIC(NMAX,YNEW,C,RANGE);
    INTEGER VALUE NMAX
    RATIONAL ARRAY VALUE RANGE
    LONG ALGEBRAIC VALUE YNEW
    LONG ALGEBRAIC ARRAY VALUE C

#   THIS PROCEDURE OUTPUTS THE RESULTS OF CGEBYSHEV SERIES
#   SOLUTION OF A LINEAR ODE.
#   INPUT PARAMETERS:
#      NMAX       -  DEGREE OF APPROXIMATION;
#      YNEW       -  THE POLYNOMIAL APPROXIMATION;
#      C          -  ARRAY C(0),...,C(NMAX) OF REQUIRED CHEBYSHEV
#                    COEFFICIENTS;
#      RANGE      -  THE FINITE INTERVAL (A,B) IN WHICH THE SOLUTION IS
#                    REQUIRED.
#   OUTPUT PARAMETERS:  NONE
#   EXTERNAL VARIABLES REQUIRED: XX.
#   PACKAGE PROCEDURES REQUIRED: CWRIT,TRANS2.
#   DECLARATION IN CALLING PROCEDURE: NONE.

    INTEGER DEGREE
    LONG ALGEBRAIC YTRANS
    EXTERNAL LONG ALGEBRAIC XX

#   OUTPUT RESULTS OF COMPUTATION.

    WRITE " THE DEGREE OF APPROXIMATION IS", NMAX
    WRITE " THE POLYNOMIAL SOLUTION IS", YNEW
    CWRIT(C,NMAX)


# TRANSFORM RESULTS BACK TO RANGE (A,B), IF NECESSARY.

    IF ( .NOT.(RANGE==(-1,1)) ) DO
       WRITE " SOLUTION TRANSFORMED BACK TO GIVEN RANGE (A,B)"
       TRANS2(RANGE,YNEW,YTRANS)
       WRITE " THE TRANSFORMED POLYNOMIAL SOLUTION IS", YTRANS
    DOEND

    RETURN
END # END OF PROCEDURE EPIC.
```

```
PROCEDURE EPIT(NMAX,YNEW,PERT,RANGE)
   INTEGER VALUE NMAX
   RATIONAL ARRAY VALUE RANGE
   LONG ALGEBRAIC VALUE PERT,YNEW

#   THIS PROCEDURE OUTPUTS THE RESULTS OF SOLUTION BY THE
#   TAU METHOD OF A LINEAR ODE.
#   INPUT PARAMETERS:
#      NMAX       -- DEGREE OF APPROXIMATION.
#      YNEW       --  THE POLYNOMIAL APPROXIMATION;
#      PERT       --  THE PERTURBATION INPOSED ON THE ORIGINAL ODE;
#      RANGE      --  THE FINITE INTERVAL (A,B) IN WHICH THE SOLUTION IS
#                     REQUIRED.
#  OUTPUT PARAMETERS:  NONE.
#  EXTERNAL VARIABLES REQUIRED: XX.
#  PACKAGE PROCEDURES REQUIRED: TCS,CHFORM,CWRIT,TRANS2.
#  DECLARATION IN CALLING PROCEDURE: NONE.

   INTEGER DEGREE
   LONG ALGEBRAIC YTRANS,PTRANS,P1
   LONG ALGEBRAIC ARRAY C
   EXTERNAL LONG ALGEBRAIC XX
   LONG ALGEBRAIC ALTRAN CHFORM
   LONG ALGEBRAIC ARRAY ALTRAN TCS

#  OUTPUT RESULTS OF COMPUTATION.

   P1 = PERT.
   C = TCS(CHFORM(YNEW,NMAX,XX),NMAX)
   DEGREE = DEG(PERT,XX)
   PERT = CHFORM(PERT,DEGREE,XX)
   WRITE " THE DEGREE OF APPROXIMATION IS", NMAX
   WRITE " THE POLYNOMIAL SOLUTION IS", YNEW
   WRITE " THE PERTURBATION IS", PERT
   CWRIT(C,NMAX)

# TRANSFORM RESULTS BACK TO RANGE (A,B), IF NECESSARY.

   IF ( .NOT.(RANGE==(-1,1)) ) DO
      WRITE " SOLUTION TRANSFORMED BACK TO GIVEN RANGE (A,B)"
      TRANS2(RANGE,YNEW,YTRANS)
      WRITE " THE TRANSFORMED POLYNOMIAL SOLUTION IS", YTRANS
      TRANS2(RANGE,P1,PTRANS)
      WRITE " THE TRANSFORMED PERTURBATION IS", PTRANS
   DOEND

   RETURN
END      # END OF PROCEDURE EPIT.
```

```
PROCEDURE EPIN(NMAX,NITER,YNEW,PERT,RANGE)
  INTEGER VALUE NMAX,NITER
  RATIONAL ARRAY VALUE RANGE
  LONG ALGEBRAIC VALUE PERT,YNEW

#   THIS PROCEDURE OUTPUTS THE RESULTS OF ITEARTIVE SOLUTION BY THE
#   TAU METHOD OF A LINEAR OR NONLINEAR ODE.
#   INPUT PARAMETERS:
#      NMAX      -  DEGREE OF APPROXIMATION;
#      NITER     -  NUMBER OF ITERATIONS PERFORMED;
#      YNEW      -  THE POLYNOMIAL APPROXIMATION;
#      PERT      -  THE PERTURBATION INPOSED ON THE ORIGINAL ODE;
#      RANGE     -  THE FINITE INTERVAL (A,B) IN WHICH THE SOLUTION IS
#                   REQUIRED.
#  OUTPUT PARAMETERS:  NONE.
#  EXTERNAL VARIABLES REQUIRED: XX.
#  PACKAGE PROCEDURES REQUIRED: TCS,CHFORM,CWRIT,TRANS2.
#  DECLARATION IN CALLING PROCEDURE: NONE.

  INTEGER DEGREE
  LONG ALGEBRAIC YTRANS,PTRANS,P1
  LONG ALGEBRAIC ARRAY C
  EXTERNAL LONG ALGEBRAIC XX
  LONG ALGEBRAIC ALTRAN CHFORM
  LONG ALGEBRAIC ARRAY ALTRAN TCS

#  OUTPUT RESULTS OF COMPUTATION.

  PAGE()
  WRITE "   %%%%%%%%%%  OUTPUT  SUMMARY  %%%%%%%%%%  "
  WRITE "  THE DEGREE OF APPROXIMATION IS", NMAX
  WRITE "  THE NUMBER OF ITERATIONS PERFORMED WAS", NITER

  P1 = PERT
  C = TCS(CHFORM(YNEW,NMAX,XX),NMAX)
  DEGREE = DEG(PERT,XX)
  PERT = CHFORM(PERT,DEGREE,XX)
  WRITE " THE POLYNOMIAL SOLUTION IS", YNEW
  WRITE " THE PERTURBATION IS", PERT
  CWRIT(C,NMAX)

# TRANSFORM RESULTS BACK TO RANGE (A,B), IF NECESSARY.

  IF ( .NOT.(RANGE==(-1,1)) ) DO
     WRITE " SOLUTION TRANSFORMED BACK TO GIVEN RANGE (A,B)"
     TRANS2(RANGE,YNEW,YTRANS)
     WRITE " THE TRANSFORMED POLYNOMIAL SOLUTION IS", YTRANS
     TRANS2(RANGE,P1,PTRANS)
     WRITE " THE TRANSFORMED PERTURBATION IS", PTRANS
  DOEND

  RETURN
 END     # END OF PROCEDURE EPIN.
```

List of Level 3 Procedures

B-36

List of Level 3 Procedures

```
PROCEDURE RECC11(ORDER,POLY,HALFN)
      INTEGER VALUE ORDER
      LONG ALGEBRAIC ARRAY VALUE POLY
      INTEGER HALFN

#  THIS PROCEDURE RETURNS THE RECURRENCE EQUATION DEFINING THE
#  CHEBYSHEV SERIES COEFFICIENTS OF A LINEAR ODE.
#  INPUT PARAMTERS:
#       ORDER    - ORDER OF ODE;
#       POLY     - POLYNOMIAL COEFFICIENTS OF ODE.
#  OUTPUT PARAMETERS:
#       HALFN    - "HALF" THE LENGHT OF THE RECURRENCE EQUATION;
#       EQN      - THE RECURRENCE EQUATION RETURNED IN THE FORM
#                  V(0)*XCR(-H) + ..... + V(2H)*XCR(H)
#                  WHERE H=HALFN, R IS AN INDETERMINATE REPRESENTED
#                  BY RR. XCR(J) REPRESENTS C(R+J) AND V(J) ARE
#                  RATIONAL EXPRESSIONS IN R.
#  EXTERNAL VARIABLES REQUIRED: MSGLVL,RR,XTR,XCR.
#  PACKAGE PROCEDURE REQUIRED: GENCHE.
#  SYSTEM PROCEDURES REQUIRED: GETBLK.
#  DECLARATION IN CALLING PROCEDURE: LONG ALGEBRAIC ALTRAN RECC11

      INTEGER I,J
      LONG ALGEBRAIC EQN,GENPOL,COEF
      EXTERNAL INTEGER MSGLVL
      EXTERNAL LONG ALGEBRAIC RR
      EXTERNAL LONG ALGEBRAIC ARRAY XTR,XCR
      LONG ALGEBRAIC ALTRAN GENCHE

      IF (MSGLVL>3) WRITE "ENTER RECC11"

#  OBTAIN THE GENERATING POLYNOMIAL GENPOL.

      GENPOL = GENCHE(POLY,ORDER,HALFN)

#   OBTAIN THE RECURRENCE EQUATION.

      EQN = 0
      DO J = -HALFN,HALFN
         COEF = GETBLK(GENPOL,XTR(J),1)
         EQN = EQN + COEF(RR=RR-J)*XCR(-J)
      DOEND

      IF (MSGLVL>3) WRITE GENPOL,EQN,HALFN, "EXIT RECC11"

      RETURN(EQN)

END # END OF PROCEDURE RECC11.
```

```
PROCEDURE RECT11(POLY,ORDER,MAXK,MINK,CODE)
   INTEGER VALUE ORDER
   INTEGER ARRAY VALUE CODE
   INTEGER MINK,MAXK
   LONG ALGEBRAIC ARRAY POLY

#  THIS PROCEDURE RETURNS THE RECURRENCE EQUATION
#  DEFINING THE POWER SERIES COEFFICIENTS OF A LINEAR
#  ODE, IN EITHER THE DIFFERENTIATED OR INTEGRATED FORM.
#  INPUT PARAMETER:
#    ORDER   -  ORDER OF ODE;
#    POLY    -  POLYNOMIAL COEFFICIENTS OF ODE;
#    CODE    -  SOLUTION CODE.
#  OUTPUT PARAMETERS:
#    MAXK,MINK - INTEGER VALUES DEFINED BELOW;
#    EQN       -  THE REQUIRED RECURRENCE EQUATION RETURNED
#                 IN THE FORM
#                 V(-H2)*XAR(-H2) + ... + V(-H1)*XAR(-H1)
#                 WHERE H1=-MINK,H2=-MAXK, R IS AN
#                 INDETERMINATE REPRESENTED AS RR,
#                 XAR(J) REPRESENTS A(R+J), AND V(J) ARE
#                 RATIONAL FUNCTIONS IN R.
#  EXTERNAL VARIABLES REQUIRED: MSGLVL,RR,XXR,XAR.
#  PACKAGE PROCEDURES REQUIRED: GENINT, GENDIF.
#  SYSTEM PROCEDURES REQUIRED: GETBLK.
#  DECLARATION IN CALLING PROCEDURE: LONG ALGEBRAIC ALTRAN RECT11.

   INTEGER I,J,MJ,K
   LONG ALGEBRAIC EQN,P,GENPOL,AJ
   EXTERNAL INTEGER MSGLVL
   EXTERNAL LONG ALGEBRAIC RR
   EXTERNAL LONG ALGEBRAIC ARRAY XXR,XAR
   LONG ALGEBRAIC ALTRAN GENINT,GENDIF

   IF (MSGLVL>3) WRITE "ENTER RECT11"

#  OBTAIN THE GENERATING POLYNOMIAL GENPOL.

   IF (CODE(3) < 4) DO       # INTEGRATED FORM.
      GENPOL = GENINT(POLY,ORDER,MAXK)
      MINK = 0
   DOEND
   ELSE DO    # DIFFERENTIATED FORM.
      GENPOL = GENDIF(POLY,ORDER,MAXK,MINK)
   DOEND

#  OBTAIN THE RECURRENCE EQUATION.

   EQN = 0
   DO J = MINK,MAXK
      AJ = GETBLK(GENPOL,XXR(J),1)
      MJ = MINK-J
      EQN = EQN + AJ(RR=RR+MJ)*XAR(MJ)
   DOEND
   MAXK = MAXK - MINK

   IF (MSGLVL>3) WRITE GENPOL,EQN,MINK,MAXK,"EXIT RECT11"
   RETURN(EQN)

END #  END OF PROCEDURE RECT11.
```

```
PROCEDURE RECT12(POLY,ORDER,H1,H2,CODE)
   INTEGER VALUE ORDER,CODE
   INTEGER H2,H1
   LONG ALGEBRAIC ARRAY POLY

#  THIS PROCEDURE RETURNS THE RECURRENCE EQUATION
#  DEFINING THE POWER SERIES COEFFICIENTS OF A LINEAR
#  ODE, IN EITHER THE DIFFERENTIATED OR INTEGRATED FORM.
#  THE RECURRENCE EQUATION FOR THE INTEGRATED FORM IS
#  WITHOUT EXPLICITLY CONVERTING THE ODE TO INTEGRATED FORM.
#  INPUT PARAMETER:
#    ORDER   -  ORDER OF ODE;
#    POLY    -  POLYNOMIAL COEFFICIENTS OF ODE;
#    CODE    -  SOLUTION CODE.
#  OUTPUT PARAMETERS:
#    MAXK,MINK - INTEGER VALUES DEFINED BELOW;
#    EQN     -  THE REQUIRED RECURRENCE EQUATION RETURNED
#               IN THE FORM
#               V(-H2)*XAR(-H2) + ... + V(-H1)*XAR(-H1)
#               WHERE H1=-MINK,H2=-MAXK, R IS AN
#               INDETERMINATE REPRESENTED AS RR,
#               XAR(J) REPRESENTS A(R+J), AND V(I) ARE
#               RATIONAL FUNCTIONS IN R.
#  EXTERNAL VARIABLES REQUIRED: MSGLVL,RR,XXR,XAR.
#  PACKAGE PROCEDURES REQUIRED: GENDIF.
#  SYSTEM PROCEDURES REQUIRED: GETBLK.
#  DECLARATION IN CALLING PROCEDURE: LONG ALGEBRAIC ALTRAN RECT11.

   INTEGER I,J,MJ,K
   LONG ALGEBRAIC EQN,P,GENPOL,AI
   EXTERNAL INTEGER MSGLVL
   EXTERNAL LONG ALGEBRAIC RR
   EXTERNAL LONG ALGEBRAIC ARRAY XXR,XAR
   LONG ALGEBRAIC ALTRAN GENDIF

   IF (MSGLVL>3) WRITE "ENTER RECT12"

#  OBTAIN THE GENERATING POLYNOMIAL GENPOL.

   GENPOL = GENDIF(POLY,ORDER,H1,H2)
```

```
#  OBTAIN THE RECURRENCE EQUATION.

   IF (CODE > 2) DO  # DIFFERENTIATED FORM.
      EQN = 0
      DO J = H2,H1
         A1 = GETBLK(GENPOL,XXR(J),1)
         EQN = EQN + A1(RR=RR-J)*XAR(-J)
         GENPOL = GENPOL - A1*XXR(J)
      DOEND
      H2 = -H2
   DOEND

   ELSE DO                    # INTEGRATED FORM.
      EQN = 0
      DO J = H2,H1
         MJ = H2-J
         A1 = GETBLK(GENPOL,XXR(J),1)
         EQN = EQN + A1(RR=RR+MJ)*XAR(MJ)
         GENPOL = GENPOL - A1*XXR(J)
      DOEND

      H1 = H1 - H2
      H2 = 0
      P = 1
      DO I = 0,ORDER-1
         P = P*(RR-I)
      DOEND

      K = ORDER - H1
      IF (K>0) P = P(RR=RR+K)

      EQN = EQN/P
   DOEND


   IF (MSGLVL>3) WRITE GENPOL,EQN,H2,H1,"EXIT RECT12"
   RETURN(EQN)

END #  END OF PROCEDURE RECT12.
```

```
PROCEDURE RECT21(POLY,ORDER,MINK,MAXK,CODE,GENPOL)
  INTEGER VALUE ORDER,CODE
  INTEGER MINK,MAXK
  LONG ALGEBRAIC GENPOL
  LONG ALGEBRAIC ARRAY POLY

# THIS PROCEDURE RETURNS THE RECURRENCE EQUATION DEFINING
# THE CANONICAL POLYNOMIALS ASSOCIATED WITH A LINEAR ODE.
# IN EITHER THE DIFFERENTIATED OR INTEGRATED FORM.
# INPUT PARAMETERS:
#   ORDER   - ORDER OF ODE;
#   POLY    - POLYNOMIAL COEFFICIENTS OF ODE;
#   CODE    - SOLUTION CODE.
# OUTPUT PARAMETERS:
#   GENPOL  -- THE GENERATING POLYNOMIAL RETURNED IN THE FORM
#              U(H1)*XXR(H1) + .... + U(H2)*XXR(H2)
#              WHERE H1=MINK,H2=MAXK,R IS AN INDETERMINATED
#              REPRESENTED AS RR, X**(R+J) IS REPRESENTED BY XXR(J),
#              AND U(I) ARE RATIONAL EXPRESSIONS IN R;
#   MINK,MAXK -DEFINED AS ABOVE;
#   EQN     - THE REQUIRED RECURRENCE EQUATION (RETURNED) IN THE
#              FORM V(H)*XQR(H) + ..... + V(-1)*XQR(-1) + V(0)*XXR(-H2),
#              WHERE H = H1-H2,Q(R+J) IS REPRESENTED BY XQR(J),AND
#              V(I) ARE RATIONAL EXPRESSIONS IN R.
# EXTERNAL VARIABLES REQUIRED: MSGLVL, RR, XX,XXR, XQR.
# PACKAGE PROCEDURES REQUIRED: GENINT, GENDIF.
# SYSTEM PROCEDURES REQUIRED: GETBLK.
# DECLARATION IN CALLING PROCEDURE: LONG ALGEBRIAC ALTRAN RECT21.

  INTEGER I,J,MJ,K,MINK
  LONG ALGEBRAIC EQN,P,AI
  EXTERNAL INTEGER MSGLVL
  EXTERNAL LONG ALGEBRAIC RR,XX
  EXTERNAL LONG ALGEBRAIC ARRAY XXR,XQR
  LONG ALGEBRAIC ALTRAN GENINT,GENDIF

  IF (MSGLVL>3) WRITE "ENTER RECT21"
# OBTAIN THE GENERATING POLYNOMIAL, GENPOL.

  IF (CODE<3) DO  #   INTEGRATED FORM.
    GENPOL = GENINT(POLY,ORDER,MAXK)
    MINK = 0
  DOEND

  ELSE DO   #  DIFFERENTIATED FORM.
    GENPOL = GENDIF(POLY,ORDER,MAXK,MINK)
  DOEND

# OBTAIN THE RECURRENCE EQUATION.

  P = 0
  DO I = MINK,MAXK-1
    AI = GETBLK(GENPOL,XXR(0),1)
    P = P + AI*XQR(I-MAXK)
  DOEND

  AI = GETBLK(GENPOL,XXR(MAXK),1)
  EQN = ((XXR(-MAXK) - P)/AI)(RR=RR-MAXK)

  IF(MSGLVL>3) WRITE GENPOL,EQN,MINK,MAXK,"EXIT RECT21"
  RETURN(EQN)
END # END OF PROCEDURE RECT21.
```

```
PROCEDURE GSC11(NMAX,ORDER,EQN,HALFN,MAXSUB,CONDN,XSUB,RHS,C)
   INTEGER VALUE NMAX,ORDER,MAXSUB,HALFN
   RATIONAL ARRAY VALUE XSUB
   LONG ALGEBRAIC ARRAY VALUE CONDN
   LONG ALGEBRAIC ARRAY C
   LONG ALGEBRAIC VALUE EQN,RHS

#  THIS PROCEDURE SOLVES THE RECURRENCE EQUATION DEFINING
#  THE CHEBYSHEV COEFFICIENTS OF THE INTEGRATED
#  FORM OF A LINEAR ODE, FOR A FINITE POLYNOMIAL APPROXIMATION.
#  THE SOLUTION IS OBTAINED BY BACKWARD RECURRENCE.
#  INPUT PARAMETERS:
#     NMAX      - DEGREE OF APPROXIMATION DESIRED;
#     ORDER     - ORDER OF ODE;
#     EQN,HALFN -- INFORMATION ABOUT RECURRENCE EQUATION;
#     MAXSUB,XSUB,CONDN - INFORMATION ABOUT ASOCIATED CONDITIONS;
#     RHS       - RIGHT-HAND-SIDE POLYNOMIAL OF ODE.
#  OUTPUT PARAMETERS:
#     C         - THE REQUIRED CHEBYSHEV COEFFICIENTS;
#     YSTAR     - THE POLYNOMIAL APPROXIMATION (RETURNED);
#  EXTERNAL VARIABLES REQUIRED: MSGLVL,XX,XPAR,RR,XCR.
#  PACKAGE PROCEDURES REQUIRED: CHFORM,VALC11,PSC11,TCS.
#  SYSTEM PROCEDURES REQUIRED: TIME,GETBLK.
#  DECLARATION IN CALLING PROCEDURE: LONG ALGEBRAIC ALTRAN GSC11.

   INTEGER I,J,M,NMIN,DEGREE,DEGRHS,N
   REAL TOLD,TNEW
   LONG ALGEBRAIC LCOEF,LCVAL,NEWEQN,YSTAR
   LONG ALGEBRAIC ARRAY C,PARLIS,PARVAL
   EXTERNAL INTEGER MSGLVL
   EXTERNAL LONG ALGEBRAIC RR,XX
   EXTERNAL LONG ALGEBRAIC ARRAY XCR,XPAR
   LONG ALGEBRAIC ALTRAN CHFORM,PSC11
   LONG ALGEBRAIC ARRAY ALTRAN TCS

   IF (MSGLVL>3) WRITE "ENTER GSC11"
   TIME(TOLD)

#   SOLVE THE RECURRENCE EQUATION FOR PARAMETRIC SOLUTION.

   YSTAR = PSC11(NMAX,NMIN,ORDER,RHS,EQN,HALFN,C)
   TNEW = TIME(TOLD)
   IF(MSGLVL>2)_
   WRITE " TIME TO COMPUTE PARAMETRIC SOLUTION WAS", TNEW
   IF (MSGLVL>3) WRITE YSTAR,RHS

#   DETERMINE THE VALUES OF THE PARAMETERS.

   VALC11(C,NMAX,ORDER,YSTAR,EQN,HALFN,HALFN+NMIN,
          MAXSUB,CONDN,XSUB,RHS,PARLIS,PARVAL)
   YSTAR = YSTAR(PARLIS=PARVAL)
   C = TCS(CHFORM(YSTAR,NMAX,XX),NMAX)
   TNEW = TIME(TOLD)
   IF (MSGLVL>2)_
   WRITE " TIME TO ASSIGN VALUES TO PARAMETERS WAS", TNEW
   IF (MSGLVL>3) WRITE YSTAR,C,PARLIS,PARVAL,"EXIT GSC11"

   RETURN(YSTAR)
END  #  END OF PROCEDURE GSC11.
```

```
PROCEDURE GSC12(NMAX,ORDER,EQN,HALFN,MAXSUB,CONDN,XSUB,RHS,C)
   INTEGER VALUE NMAX,ORDER,MAXSUB,HALFN
   RATIONAL ARRAY VALUE XSUB
   LONG ALGEBRAIC VALUE EQN,RHS
   LONG ALGEBRAIC ARRAY  C
   LONG ALGEBRAIC ARRAY VALUE CONDN

#  THIS PROCEDURE SOLVES THE RECURRENCE EQUATION DEFINING
#  THE CHEBYSHEV COEFFICIENTS OF THE INTEGRATED
#  FORM OF A LINEAR ODE, FOR A FINITE POLYNOMIAL APPROXIMATION.
#  THE SOLUTION IS OBTAINED BY GAUSSIAN ELIMINATION.
#  INPUT PARAMETERS:
#     NMAX      - DEGREE OF APPROXIMATION DESIRED;
#     ORDER     - ORDER OF ODE;
#     EQN,HALFN - INFORMATION ABOUT RECURRENCE EQUATION;
#     MAXSUB,XSUB,CONDN - INFORMATION ABOUT ASOCIATED CONDITIONS;
#     RHS       - RIGHT-HAND-SIDE POLYNOMIAL OF ODE.
#  OUTPUT PARAMETERS:
#     C         - THE REQUIRED CHEBYSHEV COEFFICIENTS;
#     YSTAR     - THE POLYNOMIAL APPROXIMATION (RETURNED);
#  EXTERNAL VARIABLES REQUIRED: MSGLVL,XX,XPAR.
#  PACKAGE PROCEDURES REQUIRED: EVAL,VALC12,CHFORM.
#  SYSTEM PROCEDURES REQUIRED: TIME,DEG,PINTN.
#  DECLARATION IN CALLING PROCEDURE: LONG ALGEBRAIC ALTRAN GSC12.

   INTEGER DEGRHS, I
   REAL TOLD,TNEW
   LONG ALGEBRAIC YSTAR
   LONG ALGEBRAIC ARRAY PARLIS,PARVAL
   EXTERNAL INTEGER MSGLVL
   EXTERNAL LONG ALGEBRAIC XX
   EXTERNAL LONG ALGEBRAIC ARRAY XPAR
   LONG ALGEBRAIC ALTRAN EVAL,CHFORM,PINTN

   IF (MSGLVL>3) WRITE " ENTER GSC12"
   TIME(TOLD)

#  OBTAIN PARAMETRIC SOLUTION.

   YSTAR = EVAL(XPAR,NMAX,XX)
   IF (RHS<>0) DO
      DEGRHS = DEG(RHS,XX) + ORDER
      RHS = CHFORM(PINTN(RHS,ORDER$XX),DEGRHS,XX)
   DOEND
   TNEW = TIME(TOLD)
   IF(MSGLVL>2) WRITE "TIME TO COMPUTE PARAMETRIC SOLUTION WAS",TNEW
   IF(MSGLVL>3) WRITE YSTAR,RHS

#  ASSIGN VALUES TO PARAMETERS.

   VALC12(NMAX,ORDER,YSTAR,EQN,HALFN,
          MAXSUB,CONDN,XSUB,RHS,PARLIS,PARVAL)
   YSTAR = YSTAR(PARLIS=PARVAL)
   C = PARVAL
   TNEW = TIME(TOLD)
   IF(MSGLVL>2)&
   WRITE "TIME TO ASSIGN VALUES TO PARAMETERS WAS", TNEW
   IF (MSGLVL>3) WRITE C,YSTAR,PARLIS,PARVAL, "EXIT GSC12"

   RETURN(YSTAR)
END  # END OF PROCEDURE GSC12.
```

B-48

```
PROCEDURE GST12(EQN,N,H1,H2,CODE,PERT,
                MAXSUB,CONDN,XSUB,RPOLY,ORDER)
    INTEGER VALUE N,ORDER,MAXSUB,H1,H2,CODE
    LONG ALGEBRAIC VALUE EQN,RPOLY
    LONG ALGEBRAIC PERT
    LONG ALGEBRAIC ARRAY VALUE CONDN
    RATIONAL ARRAY VALUE XSUB

#  THIS PROCEDURE SOLVES THE RECURRENCE EQUATION DEFINING
#  THE POWER SERIES COEFFICIENTS OF THE SOLUTION FUNCTION
#  OF A PERTURBED LINEAR ODE IN EITHER INTEGRATED
#  OR DIFFERENTIATED FORM.
#  INPUT PARAMETERS:
#     N     - DEGREE OF APPROXIMATION DESIRED;
#     ORDER - ORDER OF ODE;
#     CODE  - SOLUTION CODE;
#     EQN,H1,H2 - INFORMATION ABOUT RECURRENCE EQUATION;
#     MAXSUB,XSUB,CONDN - INFORMATION ABOUT ASSOCIATED
#                 CONDITIONS;
#     RPOLY     -  RIGHT-HAND-SIDE POLYOMIAL OF ODE.
#  OUTPUT PARAMETER:
#     YSTAR  - THE REQUIRED SOLUTION FUNCTION (RETURNED).
#     PERT   - THE PERTURBATION ON THE ODE.
#  EXTERNAL VARIABLES REQUIRED: MSGLVL,XX,XAR.
#  PACKAGE PROCEDURES REQUIRED: PST12, VALT12.
#  SYSTEM PROCEDURES REQUIRED: GETBLK, TIME.
#  DECLARATION IN CALLING PROCEDURE: LONG ALGEBRAIC ALTRAN GST12.

    INTEGER I,J,K,H,NPERT,NP
    INTEGER ARRAY H1ST
    REAL TOLD,TNEW
    LONG ALGEBRAIC YSTAR
    LONG ALGEBRAIC ARRAY (-H1:H2) COEF
    LONG ALGEBRAIC ARRAY (0:N) B
    LONG ALGEBRAIC ARRAY  PAR1IS,PARVAL,RCOEF
    EXTERNAL INTEGER MSGLVL
    EXTERNAL LONG ALGEBRAIC XX
    EXTERNAL LONG ALGEBRAIC ARRAY XAR

    IF (MSGLVL>3) WRITE "ENTER GST12"
    TIME(TOLD)

#  SET UP ARRAY OF COEFFICIENTS IN RECURRENCE EQN.

    DO I = -H1,H2
      COEF(I) = GETBLK(EQN,XAR(I),1)
    DOEND
```

```
#  OBTAIN PARAMETRIC APPROXIMATION, YSTAR, AND PERTURBATION, PERT.

    PST12(N,ORDER,H,H1,H2,NPERT,COEF,RPOLY,MAXSUB,
          CONDN,XSUB,ILIST,NP,RCOEF,B,YSTAR,PERT,CODE)
    TNEW = TIME(TOLD)
    IF (MSGLVL>2)_
    WRITE " TIME TO COMPUTE PARAMETRIC SOLUTION WAS", TNEW
    IF (MSGLVL>3) WRITE H,NP,ILIST,NPERT,RPOLY,YSTAR,PERT

#  COMPUTE THE VALUES OF THE PARAMETERS, IF ANY.

     IF ( (H+NP) > 0 ) DO
        VALT12(YSTAR,PERT,B,N,ORDER,CODE,H,NP,COEF,RCOEF,
               MAXSUB,CONDN,XSUB,ILIST,H1,H2,PARLIS,PARVAL)
        YSTAR = YSTAR(PARLIS=PARVAL)
        PERT = PERT(PARLIS=PARVAL)
     DOEND
     TNEW = TIME(TOLD)
     IF (MSGLVL>2)_
     WRITE " TIME TO ASSIGN VALUES TO PARAMETERS WAS", TNEW

     IF (MSGLVL>3) WRITE YSTAR,PERT,"EXIT GST12"
     RETURN(YSTAR)
END  #   END OF PROCEDURE GST12.
```

B-49

```
PROCEDURE GST21(N,ORDER,POLY,CODE,H1,H2,EQN,GENPOL,
                RPOLY,MAXSUB,CONDN,XSUB,PERT)
   INTEGER VALUE N,CODE,ORDER,H1,H2,MAXSUB
   RATIONAL ARRAY VALUE XSUB
   LONG ALGEBRAIC PERT
   LONG ALGEBRAIC VALUE RPOLY,EQN,GENPOL
   LONG ALGEBRAIC ARRAY VALUE CONDN,POLY

#  THIS PROCEDURE OBTAINS THE SOLUTION FUNCTION OF A PERTURBED
#  LINEAR ODE IN EITHER INTEGRATED OR DIFFERENTIATED FORM.
#  THE SOLUTION IS OBTAINED USING THE CANONICAL POLYNOMIALS
#  ASSOCIATED WITH THE LINEAR OPERATOR DEFINED BY THE ODE.
#  INPUT PARAMETER:
#      N    - DEGREE OF APPROXIMATION REQUIRED;
#      ORDER - ORDER OF ODE;
#      POLY  - POLYNOMIAL COEFFICIENTS OF ODE;
#      RPOLY - RIGHT-HAND-SIDE POLYNOMIAL OF ODE;
#      EQN,GENPOL,H1,H2 -  INFORMATION ABOUT THE RECURRENCE
#                          EQUATION;
#      MAXSUB,XSUB,CONDN - INFORMATION ABOUT THE ASSOCIATED
#                          CONDITIONS.
#  OUTPUT PARAMETERS:
#      YSTAR  - THE REQUIRED SOLUTION FUNCTION (RETURNED);
#      PERT   - PERTURBATION ON ODE.
#  EXTERNAL VARIABLES REQUIRED: MSGLVL,XX,XPAR.
#  PACAKGE PROCEDURES REQUIRED: PST21, VALT21, CANONP.
#  SYSTEM PROCEDURES REQUIRED: TIME, PINT.
#  DECLARATION IN CALLING PROCEDURE: LONG ALGEBRAIC ALTRAN GST21.

   INTEGER I,J,M,NMM,DEGREE,R,S1,V1,II
   INTEGER ARRAY S
   REAL TOLD,TNEW
   LONG ALGEBRAIC YSTAR
   LONG ALGEBRAIC ARRAY PARLIS,PARVAL,Q,V
   EXTERNAL INTEGER MSGLVL
   EXTERNAL LONG ALGEBRAIC XX
   EXTERNAL LONG ALGEBRAIC ARRAY XPAR

   IF (MSGLVL>3) WRITE " ENTER GST21"
   TIME(TOLD)

#  OBTAIN THE CANONICAL POLYNOMIALS.

   CANONP(N,POLY,ORDER,CODE,H1,H2,EQN,GENPOL,Q,H,S,S1,V,V1)

#  OBTAIN RIGHT-HAND-SIDE POLYNOMIAL.

   R = S1-V1 + ORDER
   IF (CODE<3) DO          # INTEGRATED FORM.
      R = S1-V1
      DO I = 1,ORDER
         RPOLY = PINT(RPOLY,XX) + XPAR(S1+I)
      DOEND
   DOEND
```

List of Level 4 Procedures

List of Level 4 Procedures

```
PROCEDURE GENCHE(POLY,ORDER,HALFN)
  INTEGER VALUE ORDER
  INTEGER HALFN
  LONG ALGEBRAIC ARRAY POLY

#  THIS PROCEDURE COMPUTES THE GENERATING POLYNOMIAL FOR
#    THE CHEBYSHEV SERIES SOLUTION OF THE INTEGRATED FORM
#    OF A LINEAR OF GIVEN ORDER.  THE POLYNOMIAL COEFFICIENTS
#    ARE GIVEN IN ARRAY POLY AND THE GENERATING POLYNOMIAL IS
#    RETURNED IN THE FORM
#      V(MINK)*XTR(MINK) +  ..... + V(MAXK)*XTR(MAXK)
#    WHERE R IS AN INDETERMINATE REPRESENTED AS RR,
#    XTR(J) REPRESENTS THE (R+J)TH CHEBYSHEV POLYNOMIAL,
#    AND V(I) ARE RATIONAL EXPRESSIONS IN R.
#  EXTERNAL VARIABLES REQUIRED: XX.
#  PACKAGE PROCEDURES REQUIRED: INTEGR,CHFORM,PRODTR,INFORM.
#  SYSTEM PROCEDURES REQUIRED: DEG,IMAX.
#  DECLARATION IN CALLING PROCEDURE: LONG ALGEBRAIC ALTRAN GENCHE.

  INTEGER DEGREE, I,J,M,HALF
  LONG ALGEBRAIC P,GENPOL,TERM
  EXTERNAL LONG ALGEBRAIC XX
  LONG ALGEBRAIC ALTRAN PRODTR,INTEGR,CHFORM
  LONG ALGEBRAIC ARRAY ALTRAN INFORM

  POLY = INFORM(ORDER,POLY,XX)
  HALFN = 0
  GENPOL = 0

  DO M = 0,ORDER
     P = POLY(M)
     DEGREE = DEG(P,XX)
     P = CHFORM(P,DEGREE,XX)
     TERM = PRODTR(P,DEGREE)
     HALF = DEGREE

     DO I = 1,M
        TERM = INTEGR(TERM,HALF)
        HALF = HALF + 1
     DOEND

     GENPOL = GENPOL + TERM
     HALFN = IMAX(HALFN,HALF)
  DOEND

  RETURN(GENPOL)

END # END OF PROCEDURE GENCHE.
```

```
PROCEDURE GENDIF(POLY,ORDER,MAXK,MINK)
    INTEGER VALUE ORDER
    LONG ALGEBRAIC ARRAY VALUE POLY
    INTEGER MAXK,MINK

#  THIS PROCEDURE COMPUTES THE GENERATING POLYNOMIAL FOR
#    THE POWER SERIES SOLUTION OF THE DIFFERENTIATED FORM
#    OF A LINEAR OF GIVEN ORDER.  THE POLYNOMIAL COEFFICIENTS
#    ARE GIVEN IN ARRAY POLY AND THE GENERATING POLYNOMIAL IS
#    RETURNED IN THE FORM
#      V(MINK)*XXR(MINK) +  ..... + V(MAXK)*XXR(MAXK)
#    WHERE R IS AN INDETERMINATE REPRESENTED AS RR,
#    XXR(J) REPRESENTS X**(R+J) AND V(I) ARE RATIONAL EXPRESSIONS
#    IN R.
#  EXTERNAL VARIABLES REQUIRED: XX,RR,XXR.
#  PACKAGE PROCEDURES REQUIRED: NONE.
#  SYSTEM PROCEDURES REQUIRED: DEG,GETBLK,IMAX,IMIN.
#  DECLARATION IN CALLING PROCEDURE: LONG ALGEBRAIC ALTRAN GENDIF.

    INTEGER I,J,M,MJ,DEGREE,M1,M2
    LONG ALGEBRAIC P,GENPOL,TERM,UR,AJ
    EXTERNAL LONG ALGEBRAIC XX,RR
    EXTERNAL LONG ALGEBRAIC ARRAY XXR

    UR = 1
    GENPOL = 0

    DO M = 0,ORDER
       P = POLY(M)
       IF (P<>0) DO
          DEGREE = DEG(P,XX)
          TERM = 0

          DO J = DEGREE,0,-1
             AJ = GETBLK(P,XX,J)
             IF (AJ <> 0) DO
                I = J
                TERM = TERM + AJ*XXR(-M+J)
             DOEND
          DOEND

          GENPOL = GENPOL + UR*TERM
          M1 = DEGREE-M
          M2 = I-M
          IF (NULL(MAXK)) MAXK = M1
          ELSE MAXK = IMAX(MAXK,M1)
          IF (NULL(MINK)) MINK = M2
          ELSE MINK = IMIN(MINK,M2)
       DOEND
       UR = UR *(RR-M)
    DOEND

    RETURN(GENPOL)

END # END OF PROCEDURE GENDIF.
```

```
  PROCEDURE GENINT(POLY,ORDER,MAXK)
    INTEGER VALUE ORDER
    INTEGER MAXK
    LONG ALGEBRAIC ARRAY  POLY

#  THIS PROCEDURE COMPUTES THE GENERATING POLYNOMIAL FOR
#    THE POWER SERIES SOLUTION OF THE INTEGRATED FORM
#    OF A LINEAR OF GIVEN ORDER.  THE POLYNOMIAL COEFFICIENTS
#    ARE GIVEN IN ARRAY POLY AND THE GENERATING POLYNOMIAL IS
#    RETURNED IN THE FORM
#      V(MINK)*XXR(MINK) +  ..... + V(MAXK)*XXR(MAXK)
#    WHERE R IS AN INDETERMINATE REPRESENTED AS RR,
#    XXR(J) REPRESENTS X**(R+J) AND V(I) ARE RATIONAL EXPRESSIONS
#    IN R.
#  EXTERNAL VARIABLES REQUIRED: XX.
#  PACKAGE PROCEDURES REQUIRED: INT,PRODXR,INFORM.
#  SYSTEM PROCEDURES REQUIRED: DEG,IMAX.
#  DECLARATION IN CALLING PROCEDURE: LONG ALGEBRAIC ALTRAN GENINT.

    INTEGER DEGREE,I,J,M
    LONG ALGEBRAIC P,GENPOL,TERM
    EXTERNAL LONG ALGEBRAIC XX
    LONG ALGEBRAIC ALTRAN PRODXR,INT
    LONG ALGEBRAIC ARRAY ALTRAN INFORM

    POLY = INFORM(ORDER,POLY,XX)
    MAXK = 0
    GENPOL = 0

    DO M = 0,ORDER
      # ADD APPROPRIATE TERM INTO GENPOL.
      P = POLY(M)
      DEGREE = DEG(P,XX)
      TERM = PRODXR(P,DEGREE,XX)

      DO I = 1,M
        TERM = INT(TERM,DEGREE)
        DEGREE = DEGREE + 1
      DOEND

      GENPOL = GENPOL + TERM
      MAXK = IMAX(MAXK,DEGREE)
    DOEND

    RETURN(GENPOL)
  END # END OF PROCEDURE GENINT.
```

```
PROCEDURE PSC11(NMAX,NMIN,ORDER,RHS,EQN,HALFN,C)
   INTEGER VALUE NMAX,ORDER,HALFN
   INTEGER NMIN
   LONG ALGEBRAIC ARRAY C
   LONG ALGEBRAIC VALUE RHS,EQN

#  THIS PROCEDURE RETURNS THE PARAMETRIC CHEBYSHEV SERIES SOLUTION
#    (OF DEGREE N) OF A LINEAR ODE OF GIVEN ORDER, USING
#    BACKWARD RECURRENCE.
#  EXTERNAL VARIABLES REQUIRED: RR,XX,XPAR,XCR,XT.
#  PACKAGE PROCEDURES REQUIRED: CHFORM,EVAL,BAKREC.
#  DECLARATION IN CALLING  PROCEDURE: LONG ALGEBRAIC ALTRAN PSC11.

   INTEGER DEGRHS,I,J,N
   LONG ALGEBRAIC NEWEQN,YSTAR,LCVAL,LCOEF
   LONG ALGEBRAIC ARRAY INDET,RLIST
   EXTERNAL LONG ALGEBRAIC RR,XX
   EXTERNAL LONG ALGEBRAIC ARRAY XPAR,XCR,XT
   LONG ALGEBRAIC ALTRAN CHFORM,PINTN,EVAL
   LONG ALGEBRAIC ARRAY ALTRAN BAKREC

#  COMPUTE NMIN.

   LCOEF = GETBLK(EQN,XCR(-HALFN),1)
   NMIN = 0
   DO J = 0,NMAX
      LCVAL = LCOEF(RR=J+HALFN)
      IF (LCVAL==0) NMIN = J+1
   DOEND
   WRITE NMIN

#  NOW DETERMINE THE LIST OF RIGHT-HAND-SIDES INVOLVED IN
#  SOLVING THE RECURRENCE EQUATION FOR C(NMIN),...,C(NMAX).

      DEGRHS = DEG(RHS,XX)
      IF (RHS<>0) DEGRHS = DEGRHS+ORDER
      RHS = CHFORM(PINTN(RHS,ORDER$XX),DEGRHS,XX)
      RLIST = 0$(1)
      DO I = NMIN+HALFN,DEGRHS
         RLIST = (RLIST,GETBLK(RHS,XT(I),1))
      DOEND

     N = 2*HALFN
     NEWEQN = EQN(RR = RR+HALFN)
     INDET = 1$(XCR(-HALFN))
     DO I = 1,N
        INDET = (INDET,XCR(-HALFN+I))
    DOEND

    C = BAKREC(NEWEQN,N,1$INDET,RR,RLIST,NMIN,NMAX,XPAR,HALFN)

#  INTRODUCE FURTHER PARAMETERS FOR UNDEFINED COEFFICIENTS.

   DO I = 0,NMIN-1
      C(I) = XPAR(HALFN+I+1)
   DOEND

   YSTAR = EVAL(C,NMAX,XX)
   RETURN(YSTAR)
```

```
PROCEDURE PST12(N,ORDER,H,H1,H2,NPERT,COEF,RPOLY,MAXSUB,
                CONDN,XSUB,ILIST,NP,RCOEF,B,YSTAR,PERT,CODE)
   INTEGER VALUE N,H1,H2,ORDER,MAXSUB,CODE
   INTEGER H,NP,NPERT
   INTEGER ARRAY ILIST
   RATIONAL ARRAY VALUE XSUB
   LONG ALGEBRAIC VALUE RPOLY
   LONG ALGEBRAIC ARRAY VALUE COEF,CONDN
   LONG ALGEBRAIC YSTAR,PERT
   LONG ALGEBRAIC ARRAY(0:N) B
   LONG ALGEBRAIC ARRAY RCOEF

#  THIS PROCEDURE RETURNS THE PARAMETRIC TAU METHOD SOLUTION
#    AND PERTURBATION OF A LINEAR ODE, USING DIRECT SERIES
#    SUBSTITUTION.
#  EXTERNAL VARIABLES REQUIRED: RR,XX,TPOWER,XPAR.
#  SYSTEM PROCEDURES REQUIRED: PINTN,IMAX,TPS.
#  DECLARATION IN CALLING PROCEDURE: NONE.


   INTEGER I,J,IBEG,IEND,DEGREE,JBEG,JEND
   LONG ALGEBRAIC P,SUM,LCOEF,Y1
   EXTERNAL LONG ALGEBRAIC XX,RR
   EXTERNAL LONG ALGEBRAIC ARRAY TPOWER,XPAR
   LONG ALGEBRAIC ALTRAN PINTN
   LONG ALGEBRAIC ARRAY ALTRAN TPS

#  DISTINGUISH BETWEEN DIFFERENTIATED AND INTEGRATED FORMS.

  IF ( CODE<3 ) DO          # INTEGERATED FORM.
     RPOLY = PINTN(RPOLY,ORDER$XX)
     NPERT = N
     H = H1
     IBEG = ORDER
  DOEND
  ELSE DO                   #  DIFFERENTIATED FORM.
     NPERT = N - ORDER
     H = ORDER + H1
     IBEG = 0
  DOEND

#   INITIALIZE SET OF "SPECIAL" VALUES OF I.

  IEND = H1 -1
  ILIST = 0$0
  DO I = IBEG,IEND
     ILIST = (ILIST,I)
  DOEND
#  OBTAIN PARAMETRIC FORM OF PERTURBATION.

   PERT = 0
   DO I = 1,H
      PERT = PERT + XPAR(I)*TPOWER(NPERT+I)
   DOEND
   RCOEF = TPS(RPOLY + PERT,XX,NPERT+H)
```

```
#  SET UP AND SOLVE UPPER TRIANGULAR SYSTEM DEFINING
#  THE COEFFICIENTS OF THE SOLUTION.

   NP = 0
   LCOEF = COEF(-H1)
   IBEG = H1
   IEND = N + H1
   DEGREE = N
   YSTAR = 0

   DO I = IEND,IBEG,-1
      P = LCOEF(RR=I)
      IF (P==0) DO      #  SINGULARITY IN TRIANGULAR SYSTEM.
         NP = NP + 1
         ILIST = (ILIST,I)
         B(DEGREE) = XPAR(H+NP)
      DOEND

      ELSE DO
         SUM = 0
         JBEG = -H1 + 1
         JEND = IMIN(H2,N-I)
         DO J = JBEG,JEND
            IF ( COEF(J)<>0) SUM = SUM + COEF(J)(RR=I)*(B(I+J))
         DOEND
         B(DEGREE) = (RCOEF(I) - SUM)/P
      DOEND

      YSTAR = YSTAR*XX + B(DEGREE)
      DEGREE = DEGREE - 1
   DOEND

   RETURN
END   #  END OF PROCEDURE PST12.
```

```
PROCEDURE PST21(N,RPOLY,Q,H,V,V1,S1,R,YSTAR,PERT)
   INTEGER VALUE N,H,V1,S1,R
   LONG ALGEBRAIC VALUE RPOLY
   LONG ALGEBRAIC ARRAY VALUE Q,V
   LONG ALGEBRAIC YSTAR,PERT

#  THIS PROCEDURE RETURNS THE PARAMETRIC TAU METHOD SOLUTION
#    AND PERTURBATION OF A LINEAR ODE, USING CANONICAL
#    POLYNOMIALS.
#  EXTERNAL VARIABLES REQUIRED: XX,XPAR,TPOWER.
#  SYSTEM PROCEDURES REQUIRED: TPS.
#  DECLARATION IN CALLING PROCEDURE: NONE.

   INTEGER I,J,L,K
   LONG ALGEBRAIC ARRAY B
   EXTERNAL LONG ALGEBRAIC XX
   EXTERNAL LONG ALGEBRAIC ARRAY TPOWER,XPAR
   LONG ALGEBRAIC ARRAY ALTRAN TPS

#  COMPUTE THE PERTURBATION.

   K = N+H
   L = K -(R-V1)
   PERT = 0
   DO I = 1,R
      PERT = PERT + XPAR(I)*TPOWER(L+I)
   DOEND

#  OBTAIN PARAMETRIC SOLUTION.

   RPOLY = RPOLY + PERT
   B = TPS(RPOLY,XX,K)
   YSTAR = 0
   DO I = 0,K
      YSTAR = YSTAR + B(I)*Q(I)
   DOEND

#  ADD TO YSTAR THE TERMS, IF ANY, CONTRIBUTED BY POLYNOMIAL SOLNS.

   DO I = 1,V1
      YSTAR = YSTAR + V(I)*XPAR(R+I)
   DOEND

   RETURN
END  # END OF PROCEDURE PST21.
```

```
PROCEDURE VALC11(C,N,ORDER,YSTAR,EQN,HALFN,NEQNS,
                 MAXSUB,CONDN,XSUB,RHS,PARLIS,PARVAL)
   INTEGER VALUE N,ORDER,MAXSUB,HALFN,NEQNS
   RATIONAL ARRAY VALUE XSUB
   LONG ALGEBRAIC VALUE EQN,RHS,YSTAR
   LONG ALGEBRAIC ARRAY VALUE CONDN,C

#  THIS PROCEDURE ASSIGNS VALUES TO THE PARAMETERS IN THE
#    PARAMETRIC CHEBYSHEV SERIES SOLUTION OF A GENERAL
#    LINEAR ODE, USING BACKWARD RECURRENCE.
#  EXTERNAL VARIABLES REQUIRED: RR,XX,XPAR,XCR,XT.
#  PACKAGE PROCEDURES REQUIRED: IABS,LISTS.
#  SYSTEM PROCEDURES REQUIRED: ASOLVE, GETBLK.
#  DECLARATION IN CALLING PROCEDURE: NONE.

   INTEGER I,J,M,IJ,NMI,L=IMAX(1,NEQNS)
   LONG ALGEBRAIC YSTAR,T,P,TEMP
   LONG ALGEBRAIC ARRAY(-HALFN:HALFN) COEF
   LONG ALGEBRAIC ARRAY(1:L) B=0,ROW=0
   LONG ALGEBRAIC ARRAY(1:L,1:L) MAT=0
   LONG ALGEBRAIC ARRAY PARLIS,PARVAL
   EXTERNAL LONG ALGEBRAIC XX,RR
   EXTERNAL LONG ALGEBRAIC ARRAY XPAR,XCR,XT
   INTEGER ALTRAN IABS
   LONG ALGEBRAIC ARRAY ALTRAN ASOLVE

#  SET UP SUBSTITUTION LISTS FOR ALL Y AND DERIVATIVE VALUES
#  APPEARING IN THE ASSOCIATED CONDITIONS.

   LISTS(YSTAR,ORDER,CONDN,MAXSUB,XSUB,PARLIS,PARVAL,XX)

#  SET UP THE FIRST "ORDER" ROWS OF THE LINEAR SYSTEM
#  DEFINING THE CHEBYSHEV COEFFICIENTS.

   DO I = 1,ORDER
      ROW(I) = CONDN(I)(PARLIS=PARVAL)
   DOEND

#  SET UP THE REMAINING ROWS OF THE LINEAR SYSTEM.

   DO I = -HALFN,HALFN
      COEF(I) = GETBLK(EQN,XCR(I),1)
   DOEND

   DO I = ORDER,NEQNS-1
      T = 0
      DO J = -HALFN, HALFN
         IJ = IABS(I+J)
         IF (IJ<=N) T = T + COEF(J)(RR=I)*C(IJ)
      DOEND
      P = GETBLK(RHS,XT(I),1)
      IF (I==0) P = P*2
      ROW(I+1) = T - P
    DOEND
```

```
 #  SOLVE THE SYSTEM.

    PARLIS = 0$0
    DO I = 1,NEQNS
       PARLIS = (PARLIS,XPAR(I))
       P = ROW(I)
       DO M = 1,NEQNS
          T = GETBLK(P,XPAR(M),1)
          MAT(I,M) = T
          P = P - T*XPAR(M)
        DOEND
       B(I) = - P
    DOEND

    PARVAL = ASOLVE(MAT,B,SING)
    RETURN

SING:
    IF (MAT==0 .AND. B ==0) DO
       PARVAL = NEQNS$(1)
       RETURN
    DOEND
    ELSE DO
       WRITE " *** ERROR 5031 *** "
       WRITE " UNABLE TO ASSIGN VALUES TO PARAMETERS IN SOLUTION"
       FRETURN
    DOEND

 END  #  END OF PROCEDURE VALC11.
```

```
PROCEDURE VALC12(N,ORDER,YSTAR,EQN,HALFN,
                 MAXSUB,CONDN,XSUB,RHS,PARLIS,PARVAL)
   INTEGER VALUE N,ORDER,MAXSUB,HALFN
   RATIONAL ARRAY VALUE XSUB
   LONG ALGEBRAIC VALUE EQN,RHS,YSTAR
   LONG ALGEBRAIC ARRAY PARLIS,PARVAL
   LONG ALGEBRAIC ARRAY VALUE CONDN

#  THIS PROCEDURE ASSIGNS VALUES TO THE PARAMETERS IN THE
#    PARAMETRIC CHEBYSHEV SERIES SOLUTION OF A GENERAL
#    LINEAR ODE, USING GAUSSIAN ELIMINATION.
#  EXTERNAL VARIABLES REQUIRED: RR,XX,XPAR,XCR,XT.
#  PACKAGE PROCEDURES REQUIRED: IABS,LISTS.
#  SYSTEM PROCEDURES REQUIRED: ASOLVE, GETBLK.
#  DECLARATION IN CALLING PROCEDURE: NONE.

   INTEGER I,J,M,IJ,NMI
   LONG ALGEBRAIC YSTAR,T,P,TEMP
   LONG ALGEBRAIC ARRAY(-HALFN:HALFN) COEF
   LONG ALGEBRAIC ARRAY(0:N) B=0
   LONG ALGEBRAIC ARRAY(0:N,0:N) MAT=0
   LONG ALGEBRAIC ARRAY PARLIS,PARVAL
   EXTERNAL LONG ALGEBRAIC XX,RR
   EXTERNAL LONG ALGEBRAIC ARRAY XPAR,XCR,XT
   INTEGER ALTRAN IABS
   LONG ALGEBRAIC ARRAY ALTRAN ASOLVE


#  SET UP THE FIRST "ORDER" ROWS OF THE LINEAR SYSTEM
#  DEFINING THE CHEBYSHEV COEFFICIENTS.

   DO I = 1,ORDER
   LISTS(YSTAR,ORDER,CONDN,MAXSUB,XSUB,PARLIS,PARVAL,XX)
      P = CONDN(I)(PARLIS=PARVAL)
      DO M = 0,N
         T = GETBLK(P,XPAR(M),1)
         MAT(I-1,M) = T
         P = P - T*XPAR(M)
       DOEND
      B(I-1) = - P
   DOEND

#  SET UP THE REMAINING ROWS OF THE LINEAR SYSTEM.

   DO I = -HALFN,HALFN
      COEF(I) = GETBLK(EQN,XCR(I),1)
   DOEND

   PARLIS = 0$0
   DO I = 0,N
      PARLIS = (PARLIS,XPAR(I))
   DOEND
```

```
      DO I = ORDER,N
         DO J = -HALFN, HALFN
            IJ = IABS(I+J)
            IF (IJ<= N) DO
               P = COEF(J)(RR=I)
               MAT(I,IJ) = MAT(I,IJ) + P
            DOEND
         DOEND
         P = GETBLK(RHS,XT(I),1)
         IF (I==0) P = P*2
         B(I) = P
       DOEND

#  SOLVE THE SYSTEM.


      PARVAL = ASOLVE(MAT,B,SING)
      RETURN

SING:
      IF (MAT==0 .AND. B ==0) DO
         PARVAL = N$(1)
         RETURN
      DOEND
      ELSE DO
         WRITE " *** ERROR 5031 *** "
         WRITE " UNABLE TO ASSIGN VALUES TO PARAMETERS IN SOLUTION"
         FRETURN
      DOEND

END   #  END OF PROCEDURE VALC12.
```

```
PROCEDURE VALT12(YSTAR,PERT,B,N,ORDER,CODE,H,NP,COEF,RCOEF,
                 MAXSUB,CONDN,XSUB,ILIST,H1,H2,PARLIS,PARVAL)
   INTEGER VALUE H,H1,H2,ORDER,MAXSUB,N,NP
   INTEGER ARRAY VALUE ILIST,CODE
   LONG ALGEBRAIC VALUE PERT,YSTAR
   RATIONAL ARRAY VALUE XSUB
   LONG ALGEBRAIC ARRAY VALUE COEF,CONDN,B,RCOEF
   LONG ALGEBRAIC ARRAY PARLIS,PARVAL

#  THIS PROCEDURE ASSIGNS VALUES TO THE PARAMETERS IN
#    THE PARAMETRIC TAU METHOD SOLUTION OF A GENERAL LINEAR
#    ODE, USING SERIES SUBSTITUTION.
#  EXTERNAL VARIABLES REQUIRED: RR,XXX,XPAR.
#  PACKAGE PROCEDURE REQUIRED: LISTS.
#  SYSTEM PROCEDURE REQUIRED: GETBLK,ASOLVE,IMIN.
#  DECLARATION IN CALLING PROCEDURE: NONE.

   INTEGER I,J,K,L,M,NMI,MH=IMAX(1,H+NP),JBEG,JEND
   LONG ALGEBRAIC SUM,T,TEMP
   EXTERNAL LONG ALGEBRAIC RR,XX
   EXTERNAL LONG ALGEBRAIC ARRAY XPAR
   LONG ALGEBRAIC ARRAY (1:MH) C,ROW
   LONG ALGEBRAIC ARRAY (1:MH,1:MH) MAT = 0
   LONG ALGEBRAIC ARRAY ALTRAN ASOLVE

#   SET UP SUBSTITUTION LISTS FOR ALL Y  AND DERIVATIVE VALUES
#   APPEARING IN THE ASSOCIATED CONDITIONS.

   IF (CODE ==2 .OR. CODE==4) DO        #  PERTURBED CONDITIONS.
      YSTAR = YSTAR - PERT
   DOEND
  LISTS(YSTAR,ORDER,CONDN,MAXSUB,XSUB,PARLIS,PARVAL,XX)

#   SET UP THE FIRST "ORDER" ROWS OF THE
#    LINEAR SYSTEM DEFINING THE PARAMETERS,
#   USING THE ASSOCIATED CONDITIONS.

  DO I = 1,ORDER
     ROW(I) = CONDN(I)(PARLIS=PARVAL)
  DOEND

#   SET UP THE REMAINING (IF ANY) ROWS OF THE LINEAR SYSTEM USING
#   THE SPECIAL CASES OF THE RECURRENCE EQUATION.

   L = H + NP - ORDER
   DO M = 1,L
      I = ILIST(M)
      SUM = 0
      JBEG = -IMIN(H1,I)
      JEND = IMIN(H2,N-I)
      DO J = JBEG,JEND
        IF(COEF(J)<>0) SUM = SUM + B(I+J)*COEF(J)(RR=I)
      DOEND
      ROW(ORDER+M) = SUM - RCOEF(I)
   DOEND
```

```
#   SET UP AND SOLVE COEFFICIENT MATRIX AND RIGHT HAND SIDE.

   PARLIS = 0$0
   DO K = 1,H+NP
      PARLIS = (PARLIS,XPAR(K))
      SUM = ROW(K)

      DO J = 1,H+NP
         T = GETBLK(SUM,XPAR(J),1)
         MAT(K,J) = T
         SUM = SUM - T*XPAR(J)
      DOEND
      C(K) = -SUM
   DOEND

   PARVAL = ASOLVE(MAT,C,SING)

   RETURN

#  CHECK FOR SINGULARITY IN MAT.

 SING:
   IF (MAT == 0 .AND. C == 0) DO
        PARVAL = (H+NP)$(1)
        RETURN
   DOEND
   ELSE DO
        WRITE " *** ERROR 5031 *** "
        WRITE " UNABLE TO ASSIGN VALUES TO PARAMETERS IN SOLUTION"
        FRETURN
   DOEND


END # END OF PROCEDURE VALT12.
```

```
PROCEDURE VALT21(YSTAR,PERT,CODE,S,S1,ORDER,MAXSUB,
                 CONDN,XSUB,N,PARLIS,PARVAL)
  INTEGER VALUE N,ORDER,MAXSUB,S1,CODE
  RATIONAL ARRAY VALUE XSUB
  INTEGER ARRAY VALUE S
  LONG ALGEBRAIC VALUE YSTAR,PERT
  LONG ALGEBRAIC ARRAY CONDN
  LONG ALGEBRAIC ARRAY PARLIS,PARVAL

#  THIS PROCEDURE ASSIGNS VALUES TO THE PARAMETERS IN
#    THE PARAMETRIC TAU METHOD SOLUTION OF A GENERAL LINEAR
#    ODE, USING CANONICAL POLYNOMIALS.
#  EXTERNAL VARIABLES REQUIRED: XUQ,XXX,XPAR.
#  PACKAGE PROCEDURE REQUIRED: LISTS.
#  SYSTEM PROCEDURE REQUIRED: GETBLK,ASOLVE.
#  DECLARATION IN CALLING PROCEDURE: NONE.

  INTEGER I,J,K,NPAR=S1+ORDER,M2 = IMAX(1,NPAR)
  LONG ALGEBRAIC SUM,T,TEMP,G
  LONG ALGEBRAIC ARRAY(1:M2) C,ROW
  LONG ALGEBRAIC ARRAY(1:M2,1:M2) MAT
  EXTERNAL LONG ALGEBRAIC XX
  EXTERNAL LONG ALGEBRAIC ARRAY XUQ,XPAR
  LONG ALGEBRAIC ARRAY ALTRAN ASOLVE

  IF (NPAR<1) RETURN

#  SET UP S1 ROWS OF THE LINEAR SYSTEM DEFINING THE PARAMETERS
#  USING THE COMPONENTS THE UNDEFINED CANONICAL POLYNOMIALS.

  DO I =1,S1
     T = XUQ(S(I))
     G = GETBLK(YSTAR,T,1)
     YSTAR = YSTAR - T*G
     ROW(I) = G
  DOEND

#   SET UP THE REMAINING "ORDER" ROWS OF THE LINEAR SYSTEM
#  USING THE ASSOCIATED CONDITIONS.

  IF (CODE==2 .OR. CODE==4) YSTAR = YSTAR - PERT
  LISTS(YSTAR,ORDER,CONDN,MAXSUB,XSUB,PARLIS,PARVAL,XX)

  DO I = 1,ORDER
     ROW(S1+I) = CONDN(I)(PARLIS=PARVAL)
  DOEND
```

```
#  ASSIGN COEFFICIENT MATRIX OF LINEAR SYSTEM TO MAT AND THE
#  RIGHT-HAND SIDE TO C.  SOLVE THE LINEAR SYSTEM.

  PARLIS = 0$0
  DO I = 1,NPAR
     PARLIS = (PARLIS,XPAR(I))
     SUM = ROW(I)
     DO J = 1,NPAR
        T = GETBLK(SUM,XPAR(J),1)
        MAT(I,J) = T
        SUM = SUM - T*XPAR(J)
     DOEND
     C(I) = -SUM
  DOEND
  PARVAL = ASOLVE(MAT,C,SING)
  RETURN

#  CHECK FOR SINGULARITY IN MAT.

  SING:
  IF (MAT==0 .AND. C==0) DO
     PARVAL = NPAR$(1)
     RETURN
  DOEND
  ELSE DO
     WRITE " *** ERROR 5031 *** "
     WRITE " UNABLE TO ASSIGN VALUES TO PARAMETERS IN SOLUTION"
     FRETURN
  DOEND
END #  END OF PROCEDURE VALT21.
```

List of Level 5 Procedures

List of Level 5 Procedures

```
PROCEDURE AROUND(P,N,M,X)
   INTEGER VALUE N,M
   LONG ALGEBRAIC VALUE P,X

#  THIS PROCEDURE RETURNS THE ROUNDED FORM OF A POLYNOMIAL P
#    OF GIVEN DEGREE N IN THE INDEPENDENT VARIABLE X.
#    THE ROUNDED FORM CONTAINS INTEGERS AT MOST M DIGITS LONG.
#    THE PROCEDURE ASSUMES THAT THE COEFFICIENTS OF THE POWERS
#    OF X IN P ARE RATIONAL NUMBERS.
#  PACKAGE PROCEDURES REQUIRED: QROUND.
#  SYSTEM PROCEDURES REQUIRED: TPS,TPSEVL,ANUM.
#  DECLARATION IN CALLING PROCEDURE: LONG ALGEBRAIC ALTRAN AROUND.

   LONG RATIONAL DEN
   LONG ALGEBRAIC NEWP,NUM
   LONG RATIONAL ARRAY Q
   LONG RATIONAL ARRAY ALTRAN QROUND
   LONG ALGEBRAIC ALTRAN TPSEVL
   LONG ALGEBRAIC ARRAY ALTRAN TPS

   NUM = ANUM(P,DEN)
   IF (DEN <= 10**M) RETURN(P)

   Q = TPS(P,X,N)
   Q = QROUND(Q,N,M)
   NEWP = TPSEVL(Q,X,N)

   RETURN(NEWP)
 END  # END OF PROCEDURE AROUND.
```

```
PROCEDURE ASHRND(P,N,CODE,X,MAXERR,XIND)
   INTEGER VALUE N,CODE
   REAL VALUE MAXERR
   LONG ALGEBRAIC VALUE P,X,XIND

#  THIS PROCEDURE RETURNS THE SHORTENED AND/OR ROUNDED FORM
#    (DEPENDING ON THE VALUE OF CODE) OF A POLYNOMIL, P OF
#    GIVEN DEGRRE N IN  THE INDEPENDENT VARIABLE X,
#    AND CONTAINING ONE INDETERMINATE XIND.  THE ROUNDING ERROR
#     NOTE THAT XIND MAY BE OMITTED IF NO SHORTENING IS REQUIRED.
#  PACKAGE PROCEDURES REQUIRED: AROUND,TRUNC,PREC.
#  SYSTEM PROCEDURES REQUIRED: TPS,TPSEVL,NTRM,DEG,ANUM.
#  DECLARATION IN CALLING PROCEDURE: LONG ALGEBRAIC ALTRAN ASHRND.

   INTEGER I,NTERMS,M,L,DEGN,DEGD
   LOGICAL ROUND=.FALSE., SHORT = .FALSE.
   LONG ALGEBRAIC NUM,DEN,RECDEN,NEWP
   LONG ALGEBRAIC ARRAY ANEW
   INTEGER ALTRAN PREC
   LONG ALGEBRAIC ALTRAN TRUNC,AROUND,TPSEVL
   LONG ALGEBRAIC ARRAY ALTRAN TPS

#  ANALYZE CODE.
   IF (CODE==1) RETURN(P)
   IF (CODE==2 .OR. CODE==4) ROUND = .TRUE.
   IF (CODE==3 .OR. CODE==4) SHORT = .TRUE.

#  SET UP FOR SHORTENING AND/OR ROUNDING.

   NUM = ANUM(P,DEN)
   DEGN = DEG(NUM,XIND)
   DEGD = DEG(DEN,XIND)
   NEWP = P

#  PERFORM SHORTENING AND/OR ROUNDING.

   IF (SHORT) DO                #   SHORTEN.
      IF (DEGD==0) RECDEN = 1/DEN
      ELSE RECDEN = TPSEVL(TPS(1/DEN,XIND,DEGD),XIND,DEGD)
      NEWP = TRUNC(NUM*RECDEN,N,XIND)
      DEGD = 0
   DOEND

   IF (ROUND) DO                #   ROUND.
      IF (DEGD<>0) DO
         WRITE " *** ERROR 5032 ***"
         WRITE " WRONG SOLUTION CODE SPECIFIED FOR PROBLEM"
         FRETURN
      DOEND

      NTERMS = NTRM(NEWP)
      M = PREC(MAXERR/NTERMS)
      IF ( (DEGN+DEGD) == 0) NEWP = AROUND(NEWP,N,M,X)
      ELSE DO
           ANEW = TPS(NEWP,X,N)
           DO I = 0,N
              ANEW(I) = AROUND(ANEW(I),N,M,XIND)
           DOEND
           NEWP = TPSEVL(ANEW,X,N)
      DOEND
   DOEND
   RETURN(NEWP)
```

63

```
PROCEDURE BAKREC(EQN, N, ZK, K, RLIST, KMIN, KMAX, PAR, NPAR)
  INTEGER VALUE  N, KMIN, KMAX, NPAR
  LONG ALGEBRAIC VALUE  EQN, K
  LONG ALGEBRAIC ARRAY VALUE  RLIST, PAR
  LONG ALGEBRAIC ARRAY (0:N) VALUE  ZK

  # PROCEDURE TO SOLVE A RECURRENCE EQUATION FOR AN NPAR-
  #    PARAMETER SOLUTION.
  #
  # INPUT PARAMETERS:
  #    EQN - THE LEFT SIDE OF THE RECURRENCE EQUATION;
  #    N - INDICATES THAT EQN CONTAINS N+1 TERMS;
  #    ZK - ARRAY OF THE N+1 INDETERMINATES APPEARING IN EQN
  #       REPRESENTING THE DEPENDENT VARIABLE Z -- I.E. EQN IS
  #       OF THE FORM
  #         U0 * ZK(0) + U1 * ZK(1) + . . . + UN * ZK(N)
  #       WHERE THE UI ARE RATIONAL EXPRESSIONS IN THE INDE-
  #       TERMINATE K AND ZK(I) REPRESENTS Z(K+I);
  #    K - THE NAME OF THE INDETERMINATE IN EQN;
  #    RLIST - LIST OF RIGHT-HAND SIDES OF THE RECURRENCE EQU-
  #       ATION CORRESPONDING TO THE CASES K = KMIN, ..., KMAX
  #       OF EQN(THE RIGHT-HAND SIDE IS ZERO BEYOND THE NUMBER
  #       OF ELEMENTS IN RLIST);
  #    KMIN, KMAX - INDICATE THE RANGE OF SUBSCRIPTS FOR WHICH
  #       THE SOLUTION IS TO BE COMPUTED -- I.E. THE SOLUTION
  #       Z(KMIN), ..., Z(KMAX) IS DESIRED;
  #    PAR - NAMES OF THE PARAMETERS TO BE USED;
  #    NPAR - NUMBER OF PARAMETERS DESIRED IN THE SOLUTION.
  #
  # OUTPUT:
  #     THE VALUE RETURNED IS AN ARRAY DIMENSIONED FROM 0 TO
  # KMAX CONTAINING THE DESIRED NPAR-PARAMETER SOLUTION.  IF
  # KMIN > 0 THEN THE FIRST KMIN ELEMENTS OF THE ARRAY ARE
  # ARBITRARILY SET TO ZERO.
  #
  # ASSUMPTIONS:
  #    IT IS ASSUMED THAT KMIN >= 0, KMAX >= KMIN+NPAR, AND
  # THAT KMAX >= KMIN+LENGTH-1 WHERE LENGTH IS THE LENGTH OF
  # THE ARRAY RLIST.
  #
  #    THE FOLLOWING DECLARATION MUST APPEAR IN THE CALLING
  # PROCEDURE:
  #          LONG ALGEBRAIC ARRAY ALTRAN BAKREC .


  INTEGER  I, J, NZEROS, KVAL, KLAST, LENGTH, L
  LONG ALGEBRAIC  U0, ZK0
  LONG ALGEBRAIC ARRAY  UNLIST
  LONG ALGEBRAIC  ZKVAL
  LONG ALGEBRAIC ARRAY  GENSOL, SOLN, VALIST
  LONG ALGEBRAIC ARRAY (0:KMAX)  Z
```

```
# SOLVE RECURRENCE FOR Z(K).

U0 = GETBLK(EQN, ZK(0), 1)
ZK0  =  ZK(0)  -  EQN / U0


# 'UNLIST' IS THE LIST OF UNKNOWNS IN ZK0.

UNLIST = 1$( ZK(1) )
DO L = 2, N
   UNLIST = (UNLIST, ZK(L))
DOEND


# COMPUTE NPAR INDEPENDENT SOLUTIONS AND FORM THEIR LINEAR
#    COMBINATION INTO GENSOL.

GENSOL = 0
DO I = 1, NPAR

   SOLN = 1$(1)
   NZEROS = N

   DO KVAL = KMAX-I, KMIN, -1

      NZEROS = NZEROS - 1
      IF (NZEROS >= 0)  VALIST = (SOLN, NZEROS$0)
         ELSE DO
              VALIST = 1$( SOLN(1) )
              DO J = 2, N
                 VALIST = (VALIST, SOLN(J))
              DOEND
             DOEND  # END OF ELSE-CLAUSE.

      ZKVAL  =  ZK0(K = KVAL)(UNLIST = VALIST)
      SOLN = (ZKVAL, SOLN)

   DOEND

   GENSOL  =  GENSOL  +  PAR(I) * (SOLN, (I-1)$0)

DOEND
```

```
   # COMPUTE A PARTICULAR SOLUTION AND ADD IT INTO GENSOL.

   LENGTH = DBINFO(RLIST)(0,1)

   IF (LENGTH > 0)  DO

      KLAST  =  KMIN + LENGTH - 1
      SOLN  =  1$( RLIST(LENGTH) / U0(K=KLAST) )
      NZEROS = N;  L = LENGTH

      DO KVAL = KLAST-1, KMIN, -1

         NZEROS = NZEROS - 1
         IF (NZEROS >= 0)  VALIST = (SOLN, NZEROS$0)
            ELSE DO
                 VALIST = 1$( SOLN(1) )
                 DO J = 2, N
                    VALIST = (VALIST, SOLN(J))
                 DOEND
                DOEND  # END OF ELSE-CLAUSE.

         ZKVAL  =  ZK0(K = KVAL)(UNLIST = VALIST)
         L  =  L - 1
         ZKVAL  =  ZKVAL  +  RLIST(L) / U0(K=KVAL)
         SOLN = (ZKVAL, SOLN)

      DOEND

      GENSOL  =  GENSOL  +  (SOLN, (KMAX-KLAST)$0)

   DOEND  # END OF IF-STATEMENT.


   # RETURN 'GENSOL' IN POSITIONS KMIN TO KMAX OF ARRAY Z,
   #    WHICH IS DIMENSIONED FROM 0 TO KMAX.

   Z = 1$(KMIN$0, GENSOL)

   RETURN( Z )

END  # END OF PROCEDURE BAKREC .
```

```
PROCEDURE CANONP(N,POLY,ORDER,CODE,H1,H2,
                 EQN,GENPOL,Q,H,S,S1,V,V1)
    INTEGER VALUE N,ORDER,CODE,H1,H2
    LONG ALGEBRAIC VALUE EQN,GENPOL
    INTEGER H=H2,S1,V1
    INTEGER ARRAY S
    LONG ALGEBRAIC ARRAY POLY,Q,V

#   THIS PROCEDURE COMPUTES THE CANONICAL POLYNOMIALS ASSOCIATED WITH
#   A LINEAR OPERATOR WITH POLYNOMIAL COEFFICIENTS.
#   INPUT PARAMETERS:
#      N     - MAXIMUM DEGREE (IN THE INDEPENDENT VARIABLE X) OF
#              THE CANONICAL POLYNOMIALS;
#      ORDER - ORDER OF LINEAR OPERATOR;
#      POLY  - ARRAY POLY(0),..,POLY(ORDER) OF POLYNOMIAL COEFFICIENTS
#              OF LINEAR OPERATOR;
#      CODE  - SOLUTION CODE INDICATING THE FORM (INTEGRATED OR
#              DIFFERENTIATED) IN WHICH THE OPERATOR IS APPLIED;
#      EQN   - THE RECURRENCE EQUATION DEFINING THE CANONICAL
#              POLYNOMIALS;
#      GENPOL - THE GENERATING POLYNOMIAL ASSOCIATED WITH THE
#               THE LINEAR OPERATOR, IN THE FORM
#               V(H1)*XXR(H1) + ... + V(H2)*XXR(H2),
#               WHERE R IS AN INDETERMINATE REPRESENTED AS RR, XXR(J)
#               REPRESENTS X**(R+J), AND V(I) ARE RATIONAL EXPRESSIONS IN R.
#   OUTPUT PARAMETERS:
#      S     - MINIMAL SET OF INDICES OF UNDEFINED CANONICAL POLYNOMIALS;
#      V     - SET OF POLYNOMIAL SOLUTIONS OF THE HOMOGENEOUS FORM
#              OF THE LINEAR OPERATOR;
#      S1,V1 - CARDINALITY OF S AND V RESPECTIVELY;
#      Q     - ARRAY Q(0),...,Q(N+H) OF CANONICAL POLYNOMIALS;
#      H     - THE "HEIGHT" OF THE LINEAR OPERATOR.
#   EXTERNAL VARIABLES REQUIRED: MSGLVL,XX,RR,XXR.
#   PACKAGE PROCEDURES REQUIRED: RECT21,GENT,EVALG,REDUZS.
#   DECLARATION IN CALLING PROCEDURE: NONE.

    INTEGER I,J,MM,Z1,G1
    INTEGER ARRAY Z,G
    LONG ALGEBRAIC XP
    EXTERNAL INTEGER MSGLVL
    EXTERNAL LONG ALGEBRAIC XX,RR
    EXTERNAL LONG ALGEBRAIC ARRAY XXR
    LONG ALGEBRAIC ALTRAN RECT21
    LONG ALGEBRAIC ARRAY ALTRAN GENTQ

    IF(MSGLVL>3) WRITE "ENTER CANONP"

#   COMPUTE (TENTATIVE) Q AND THE SET Z OF THE INDICES OF
#   CANONICAL POLYNOMIALS THAT CANNOT BE GENERATED FROM THE
#   RECURRENCE EQUATION.
#   REDUCE THE SET Z TO S, THE MINIMAL SET OF INDICES OF UNDEFINED
#   CANONICAL POLYNOMIALS.  OBTAIN POLYNOMIAL SOLUTIONS, IF ANY.

    Q = GENTQ(N,H1,H2,EQN,Z,Z1)
    EVALG(N,H,GENPOL,G,G1)
    REDUZS(Q,Z,Z1,G,G1,H,N,GENPOL,V,V1,S,S1)
    IF(MSGLVL>3) WRITE Q,Z,S,G,V,H,GENPOL,"EXIT CANONP"
    RETURN
END #  END OF PROCEDURE CANONP.
```

```
PROCEDURE ERREST(ORDER,PERT,POLY,CODE,MAXERR)
  INTEGER VALUE ORDER,CODE
  REAL MAXERR
  LONG ALGEBRAIC VALUE PERT
  LONG ALGEBRAIC ARRAY VALUE POLY

# THIS PROCEDURE RETURNS AN ESTIMATE ERR, (TOGETHER
#   WITH ITS MAXIMUM VALUE, MAXERR) OF THE ERROR IN A TAU METHOD
#   APPROXIMATE SOLUTION OF A LINEAR ODE, USING THE PERTURBATION,
#   PERT, IMPOSED ON THE LINEAR ODE.
# EXTERNAL VARIABLES REQUIRED: XX,MAXMU,XMU,MSGLVL.
# PACKAGE PROCEDURES REQUIRED: AMAX, TRUNC.
# SYSTEM PROCEDURES REQUIRED: DEG, SR.
# DECLARATION IN CALLING PROCEDURE: LONG ALGEBRAIC ALTRAN ERREST.

  INTEGER I,DEGREE,MM
  LONG ALGEBRAIC ERR,ERRSOL,P,Q
  EXTERNAL INTEGER MAXMU,MSGLVL
  EXTERNAL LONG ALGEBRAIC ARRAY XMU
  EXTERNAL LONG ALGEBRAIC XX
  LONG ALGEBRAIC ALTRAN AMAX
  LONG ALGEBRAIC ALTRAN TRUNC

  IF (MSGLVL>3) WRITE " ENTER ERREST"

  DEGREE = DEG(PERT,XX)
  P = PERT
  Q = POLY(ORDER)
  ERR = P/Q
  MM = IMAX(1,MAXMU)
  P = P(XMU=MM$(1))
  IF (DEG(Q,XX)>0) Q = TRUNC(Q,0,XX)
  P = P/Q(XMU=MM$(0))
  ERRSOL = AMAX(P,DEGREE,XX)
  MAXERR = SR(ERRSOL)

  IF (MSGLVL>3) WRITE ERR,MAXERR, " EXIT ERREST"
  RETURN(ERR)

END # END OF PROCEDURE ERREST.
```

```
PROCEDURE EVALG(N,H,GENPOL,G,G1)
  INTEGER VALUE N,H
  LONG ALGEBRAIC VALUE GENPOL
  INTEGER G1
  INTEGER ARRAY G

#  THIS PROCEDURE EVALUATES THE SET OF INADEQUATE INDICES G,
#  ASSOCIATED WITH THE GENERATING POLYNOMIAL OF A LINEAR ODE,
#  WITH HEIGHT H.  G1 IS THE CARDINALITY OF G.
#  EXTERNAL VARIABLES REQUIRED: XX,RR,XXR.
#  SYSTEM PROCEDURES REQUIRED: DEG.
#  DECLARATION IN CALLING PROCEDURE: NONE.

  INTEGER NPH,I,J
  LONG ALGEBRAIC F,XI
  LONG ALGEBRAIC ARRAY XLIST,VALIST
  EXTERNAL LONG ALGEBRAIC XX,RR
  EXTERNAL LONG ALGEBRAIC ARRAY XXR

  NPH = N+H
  DO I = 0,H
    XLIST = (XLIST,XXR(I))
  DOEND

  G = 0$0
  G1 = 0
  DO I = 0,NPH
    XI = XX**I
    VALIST = 0$0
    DO  J=0,H
        VALIST = (VALIST,XI)
        XI = XI*XX
    DOEND
    F = GENPOL(RR=I)(XLIST = VALIST)

    IF (DEG(F,XX) < I+H) DO
        G = (G,I)
        G1 = G1 + 1
    DOEND
  DOEND

END  #  END OF PROCEDURE EVALG.
```

```
PROCEDURE GENTQ(N,H1,H2,EQN,Z,Z1)
   INTEGER VALUE N,H1,H2
   INTEGER Z1
   INTEGER ARRAY Z
   LONG ALGEBRAIC VALUE EQN

#  THIS PROCEDURE GENERATES THE TENTATIVE CANONICAL POLYNOMIALS, Q,
#    ASSOCIATED WITH A LINEAR OPERATOR, GIVEN THE RECURRENCE
#    EQUATION DEFINING THE CANONICAL POLYNOMIALS.  THE PROCEDURE ALSO
#    THE SET Z OF INDICES OF THE CANONICAL POLYNOMIALS
#    THAT CANNOT BE GENERATED FROM THE RECURRENCE EQUATION.
#    Z1 IS THE CARDINALITY OF Z.
#  EXTERNAL VARIABLES REQUIRED: XX,RR,XXR,XQR,XUQ.
#  SYSTEM PROCEDURES REQUIRED: ANUM,GETBLK,IMAX.
#  DECLARATION IN CALLING PROCEDURE: LONG ALGEBRAIC ARRAY ALTRAN GENTQ.

   INTEGER I,HH=H1-H2,NH=N+H2,M1,J
   LONG ALGEBRAIC XP,DEN,DI,P
   LONG ALGEBRAIC ARRAY(0:NH) Q
   LONG ALGEBRAIC ARRAY(HH:0) COEF
   EXTERNAL LONG ALGEBRAIC XX,RR
   EXTERNAL LONG ALGEBRAIC ARRAY XQR,XUQ,XXR

#  SET UP THE COEFFICIENTS IN RECURRENCE EQUATION.

   EQN = ANUM(EQN,DEN)
   DO I = HH,0
      COEF(I) = GETBLK(EQN,XQR(I),1)
   DOEND
   XP = GETBLK(EQN,XXR(-H2),1)

#  INITIALIZE SET Z.

   Z=0$0
   DO I = 0,H2-1
      Q(I) = XUQ(I)
      Z = (Z,I)
   DOEND
   Z1 = H2
```

```
# COMPUTE THE REQUIRED CANONICAL POLYNOMIALS.

   DO I = H2,NH
      DI = DEN(RR=I)
      IF (DI <> 0) DO                  # Q(I) DEFINED.
         Q(I) = XP(RR=I)
         M1 = IMAX(HH,-I)
         DO J = M1,-1
            P = COEF(J)(RR=I)
            Q(I) = Q(I) + P*Q(I+J)
         DOEND
         Q(I) = Q(I)/DI
      DOEND

      ELSE DO                          # Q(I) UNDEFINED.
         Q(I) = XUQ(I)
         Z = (Z,I)
         Z1 = Z1 + 1
      DOEND
      XP = XP*XX
   DOEND

   RETURN(Q)
END  # END OF PROCEDURE GENTQ.
```

```
PROCEDURE INIT(ORDER,N,MAXSUB,XSUB,CONDN,X)
    INTEGER VALUE ORDER,N,MAXSUB
    RATIONAL ARRAY VALUE XSUB
    LONG ALGEBRAIC VALUE X
    LONG ALGEBRAIC ARRAY VALUE CONDN

#  THIS PROCEDURE COMPUTES AN INITIAL POLYNOMIAL APPROXIMATION,
#    YINIT, SATISFYING THE AASSOCIATED CONDITIONS OF AN ODE.
#    AN ERROR IS REPORTED IF NO INITIAL CAN BE COMPUTED WITH
#    DEGREE NOT EXCEEDING N.
#  EXTERNAL VARIABLES REQUIRED: MSGLVL,XPAR.
#  SYSTEM PROCEDURES REQUIRED: ASOLVE,TPSEVL,GETBLK,LISTS.
#  DECLARATION IN CALLING PROCEDURE: LONG ALGEBRAIC ALTRAN INIT.

    INTEGER I,J,K,DEGP,ORDM1=ORDER-1
    LONG ALGEBRAIC T,TEMP,XP,P,YINIT=0,P0,XI,XIM1
    LONG ALGEBRAIC ARRAY (0:ORDM1) C=0
    LONG ALGEBRAIC ARRAY (0:ORDM1,0:ORDM1) MAT=0
    LONG ALGEBRAIC ARRAY PARLIS,UNLIST,VALIST
    LONG ALGEBRAIC ALTRAN TPSEVL
    LONG ALGEBRAIC ARRAY ALTRAN ASOLVE
    EXTERNAL INTEGER MSGLVL
    EXTERNAL LONG ALGEBRAIC ARRAY XPAR

    IF(MSGLVL>3) WRITE "ENTER INIT"
    P0 = TPSEVL(XPAR,X,ORDM1)
    PARLIS = 0$0
    DO I = 0,ORDM1
       PARLIS = (PARLIS,XPAR(I))
    DOEND

    P = P0
    DEGP = ORDM1
    XI = X
    XIM1 = 1

AGAIN:  LISTS(P,ORDER,CONDN,MAXSUB,XSUB,UNLIST,VALIST,X)
    IF(MSGLVL>3) WRITE P,UNLIST,VALIST

#   SET UP COEFFICIENT MATRIX OF LINEAR SYSTEM DEFINING
#   THE INITIAL APPROXIMATION, YINIT.

   DO I = 1,ORDER
      XP = CONDN(I)(UNLIST=VALIST)
      TEMP = 0
      DO J = 0,ORDM1
         T = GETBLK(XP,XPAR(J),1)
         MAT(I-1,J) = T
         TEMP = TEMP + T*XPAR(J)
       DOEND
      C(I-1) = TEMP - XP
   DOEND
   IF(MSGLVL>3) WRITE MAT,C
```

```
# SOLVE LINEAR SYSTEM , IF POSSIBLE, FOR YINIT.

    C = ASOLVE(MAT,C,SING)
    IF (.NOT. (C==0) .OR. DEGP==N) DO
       YINIT = P(PARLIS = C)
       IF (MSGLVL>3) WRITE YINIT, "EXIT INIT"
       RETURN(YINIT)
    DOEND

 # TRY A NEW APPROXIMATION.

NEXTP:
    P = P0*XI + XIM1
    XIM1 = XIM1 + XI
    XI = XI*X
    GO TO AGAIN

 # RESOLVE SINGULARITY HERE.

SING:
    IF(MSGLVL>3) WRITE "SINGULAR"
    IF (DEGP > N) DO
       WRITE "  ******* ERROR 5031  *****"
       WRITE " UNABLE TO COMPUTE INITIAL APPROXIMATION "
       FRETURN
    DOEND

 END # END OF PROCEDURE INIT.
```

```
PROCEDURE INPCON(ORDER,MAXSUB,XSUB,CONDN)
   INTEGER VALUE ORDER
   INTEGER MAXSUB
   RATIONAL ARRAY XSUB
   EXTERNAL INTEGER MAXORD,MAXMU,MAXVAL
   LONG ALGEBRAIC(DY(0:MAXORD),Y,X,MU(1:IMAX(1,MAXMU)),
            DYX(1:MAXORD,1:MAXVAL),YX(1:MAXVAL)) CONDN,P
   ARRAY CONDN

#  THIS PROCEDURE READS IN THE CONDITIONS ASSOCIATED WITH
#    AN ODE OF GIVEN ORDER.  THE CONDITIONS ARE RETURNED IN
#    CONDN(1),...,CONDN(ORDER), AND THE X-VALUES IN
#    XSUB(1),...,XSUB(MAXSUB), WHERE MAXSUB IS THE NUMBER
#    OF X-VALUES APPERAING IN THE CONDITIONS.
#    THE PROCEDURE ASSUMES THAT ALL X-VALUES APPEARING IN THE
#    CONDITIONS ARE ASSIGNED VALUES.
#  EXTERNAL VARIABLES REQUIRED: MAXORD,MAXMU,MAXVAL.
#  SYSTEM PROCEDURES REQUIRED: DEG.
#  DECLARATION IN CALLING PROCEDURE: NONE.

   INTEGER I,J,K,M
   RATIONAL R
   LONG ALGEBRAIC ARRAY(1:IMAX(1,ORDER)) CD=P

   MAXSUB = 0
   IF (ORDER<1) RETURN

#   READ IN ASSOIATED CONDITIONS.

   DO M = 1,ORDER; READ CD(M); DOEND
   CONDN = CD
   DO M = 1,ORDER; WRITE CONDN(M); DOEND

#   SET UP TO READ IN XVALUES APPEARING IN ASSOC. CONDITIONS.

   DO J = MAXVAL,1,-1
      DO M = 1,ORDER
         IF (DEG(CONDN(M),YX(J))>0) GO TO DEF
         DO I = 1,ORDER-1
            IF (DEG(CONDN(M),DYX(I,J)) > 0) GO TO DEF
         DOEND
      DOEND
   DOEND
DEF:  MAXSUB = J

    XSUB = 0$0
    DO J = 1,MAXSUB
       READ R
       XSUB = (XSUB,R)
       WRITE XSUB(J)
    DOEND

   RETURN
END #  END OF PROCEDURE INPCON.
```

```
PROCEDURE INPODE(ORDER,DIFFEQ,LINEAR)
   INTEGER VALUE ORDER
   LOGICAL LINEAR
   EXTERNAL INTEGER MAXMU,MAXORD
   LONG ALGEBRAIC(DY(0:MAXORD),Y,X,MU(1:IMAX(1,MAXMU))) DIFFEQ

#  THIS PROCEDURE READS IN AN ODE OF GIVEN ORDER.
#    THE ODE IS REPRESENTED AS DIFFEQ=0.
#    THE LOGICAL VARIABLE LINEAR IS SET TO .TRUE. IF
#    THE ODE IS LINEAR AND TO .FALSE. IF OTHERWISE.
#  EXTERNAL VARIABLES REQUIRED: MAXMU,MAXORD
#  SYSTEM PROCEDURES REQUIRED: GETBLK,DEG.
#  DECLARATION IN CALLING PROCEDURE: NONE.

   INTEGER I,J,K,L,M
   LONG ALGEBRAIC P,FCHECK

#   READ IN ODE.

   READ DIFFEQ
   WRITE DIFFEQ

#   CHECK IF ODE IS LINEAR.

    LINEAR = .FALSE.
    FCHECK = DIFFEQ(Y=DY(0))
    DO I = 0,ORDER
       L = DEG(FCHECK,DY(I))
       IF (L==0) GO TO ON1
       IF (L > 1) GO TO ON2
       P = GETBLK(FCHECK,DY(I),1)
       DO J = I+1,ORDER
          IF (DEG(P,DY(I)) <> 0) GO TO ON2
       DOEND

ON1: DOEND
   LINEAR = .TRUE.
ON2: RETURN
END  #  END OF PROCEDURE INPODE.
```

```
PROCEDURE INT(P,DEGREE)
   INTEGER VALUE DEGREE
   LONG ALGEBRAIC VALUE P

#  THIS PROCEDURE INTEGRATES A POLYNOMIAL P WHICH IS ASSUMED
#    TO BE IN POWER SERIES FORM
#      V(0)*XXR(0) + ..... + V(H)*XXR(H)
#    WHERE H=DEGREE, R IS AN INDETERMINATE REPRESENTED AS
#    RR, XXR(J) REPRESENTS X**(R+J), AND V(I) ARE RATIONAL
#    EXPRESSIONS IN R.
#  EXTERNAL VARIABLES REQUIRED: RR,XXR.
#  SYSTEM PROCEDURES REQUIRED: GETBLK.
#  DECLARATION IN CALLING PROCEDURE: LONG ALGEBRAIC ALTRAN INT.

   INTEGER J
   LONG ALGEBRAIC NEWP
   EXTERNAL LONG ALGEBRAIC RR
   EXTERNAL LONG ALGEBRAIC ARRAY XXR

   NEWP = 0
   DO J = 0,DEGREE
      NEWP = NEWP + GETBLK(P,XXR(J),1)*XXR(J+1)/(RR+J+1)
   DOEND

   RETURN(NEWP)

END  #  END OF PROCEDURE INT.
```

```
PROCEDURE INTEGR(P,HALFN)
   INTEGER VALUE HALFN
   LONG ALGEBRAIC VALUE P

#  THIS PROCEDURE INTEGRATES A POLYNOMIAL P WHICH IS ASSUMED
#    TO BE IN THE CHEBYSHEV FORM
#     V(0)*XTR(-H) + .... + V(H)*XTR(H)
#    WHERE H=HALFN, R IS AN INDETERMINATE REPRESENTED AS RR,
#    XTR(J) REPRESENTS THE (R+J)-TH CHEBYSHEV POLYNOMIAL,
#    AND V(I) ARE RATIONAL EXPRESSIONS IN R.
#  EXTERNAL VARIABLES REQUIRED: RR,XTR.
#  SYSTEM PROCEDURES REQUIRED: GETBLK.
#  DECLARATION IN CALLING PROCEDURE: LONG ALGEBRAIC ALTRAN INTEGR.

   INTEGER J
   LONG ALGEBRAIC NEWP
   EXTERNAL LONG ALGEBRAIC RR
   EXTERNAL LONG ALGEBRAIC ARRAY XTR

   NEWP = 0
   DO J = -HALFN,HALFN
      NEWP = NEWP + GETBLK(P,XTR(J),1)* (XTR(J+1)/
                (RR+J+1) - XTR(J-1)/(RR+J-1))/2
   DOEND

   RETURN(NEWP)

END #  END OF PROCEDURE INTEGR.
```

```
PROCEDURE NEXTIT(N,NMAX,ORDER,YNEW,YCORR,POLY,PERT,CODE,
                 DIFFEQ,RHS)
   INTEGER VALUE ORDER,CODE,NMAX
   INTEGER N
   LONG ALGEBRAIC RHS,YNEW,PERT
   LONG ALGEBRAIC ARRAY POLY
   LONG ALGEBRAIC VALUE YCORR,DIFFEQ

#  THIS PRCEDURE SETS UP FRO THE NEXT STAGE IN THE
#    (NEWTON) ITERATIVE SOLUTION OF AN ODE.
#  EXTERNAL VARIABLES REQUIRED: XX,XY,MASGLVL,XMU,XDY.
#  PACKAGE PROCEDURES REQUIRED: ERREST,ASHRND,SETP2.
#  SYSTEM PROCEDURES REQUIRED: IMIN,DIFF.
#  DECLARATION IN CALLING PROCEDURE: NONE.

   INTEGER I,M,M1,M2,MAXN,DEGREE
   REAL MAXERR
   LONG ALGEBRAIC ERR,P
   LONG ALGEBRAIC ARRAY UNLIST,VALIST
   EXTERNAL INTEGER MSGLVL
   EXTERNAL LONG ALGEBRAIC XX,XY
   EXTERNAL LONG ALGEBRAIC ARRAY XMU,XDY
   LONG ALGEBRAIC ALTRAN ERREST,ASHRND
   LONG ALGEBRAIC ARRAY ALTRAN SETP2

   YNEW = YNEW + YCORR
   IF (CODE > 1 .AND. YCORR<>0) DO
      IF (CODE==2 .OR. CODE==4) DO
         ERR = ERREST(ORDER,PERT,POLY,CODE,MAXERR)
      DOEND

      YNEW = ASHRND(YNEW,N,CODE,XX,MAXERR,XMU(1))
   DOEND

   IF (MSGLVL>1) OUTIT(N,POLY,RHS,YNEW,YCORR,PERT)

   IF (N < NMAX) DO        #  SET UP FOR NEXT ITERATION.
      MAXN = N
      N = IMIN(N*2,NMAX)
      POLY = SETP2(N,ORDER,MAXN,DIFFEQ,YNEW,RHS)
   DOEND

   ELSE DO                 #  THIS WAS THE LAST ITERATION.

      #  OBTAIN PERTURBATION ON ORIGINAL SYSTEM.

      UNLIST = 1$(XY)
      VALIST = 1$(YNEW)
      P = YNEW
      DO I = 1,ORDER
         UNLIST = (UNLIST,XDY(I))
         P = DIFF(P,XX)
         VALIST = (VALIST,P)
      DOEND
      PERT = DIFFEQ(UNLIST=VALIST)

   DOEND

   RETURN
END # END OF PROCEDURE NEXTIT.
```

```
PROCEDURE OUTIT(N,POLY,RHS,YNEW,YCORR,PERT)
   INTEGER VALUE N
   LONG ALGEBRAIC VALUE RHS,YNEW,YCORR,PERT
   LONG ALGEBRAIC ARRAY VALUE POLY

#  THIS PROCEDURE OUTPUTS THE RESULTS OF THE CURRENT ITERATION
#    IN THE SOLUTION OF AN ODE.
#  EXTERNAL VARIABLES REQUIRED: XX.
#  PACKAGE PROCEDURES REQUIRED: CHFORM, TCS, CWRIT.
#  SYSTEM PROCEDURES REQUIRED: DEG.
#  DECLARATION IN CALLING PROCEDURE: NONE.

   INTEGER DEGREE
   LONG ALGEBRAIC ARRAY C
   EXTERNAL LONG ALGEBRAIC XX
   LONG ALGEBRAIC ALTRAN CHFORM
   LONG ALGEBRAIC ARRAY ALTRAN TCS

   WRITE " THE COEFICIENTS OF THE ODE ARE", POLY
   WRITE " THE RIGHT HAND SIDE OF ODE IS", RHS
   WRITE " THE DEGREE OF APPROXIMATION IS", N
   WRITE " THE CORRECTION TERM IS", YCORR
   WRITE "THE POLYNOMIAL SOLUTION IS", YNEW
   C = TCS(CHFORM(YNEW,N,XX),N)
   CWRIT(C,N)

   DEGREE = DEG(PERT,XX)
   PERT = CHFORM(PERT,DEGREE,XX)
   WRITE "THE PERTUBATION IS", PERT

   RETURN
END  #  END OF PROCEDURE OUTIT.
```

```
PROCEDURE PRODTR(P,DEGREE)
   INTEGER VALUE DEGREE
   LONG ALGEBRAIC VALUE P

#  THIS PROCEDURE MULTIPLIES A POLYNOMIAL P OF GIVEN DEGREE
#    ASSUMED TO BE IN CHEBYSHEV FORM, BY THE CHEBYSHEV
#    POLYNOMIAL OF DEGREE R WHERE R IS AN INDETERMINATE.
#  EXTERNAL VARIABLES REQUIRED: XTR,XT.
#  SYSTEM PROCEDURES REQUIRED: GETBLK.
#  DECLARATION IN CALLING  PROCEDURE: LONG ALGEBRAIC ALTRAN PRODTR.

   INTEGER J
   LONG ALGEBRAIC COEF,NEWP
   EXTERNAL LONG ALGEBRAIC ARRAY XTR,XT

   NEWP = 0
   DO J = 0,DEGREE
      COEF = GETBLK(P,XT(J),1)
      NEWP = NEWP + COEF*(XTR(J)+XTR(-J))/2
   DOEND

   RETURN(NEWP)

END  # END OF PROCEDURE PRODTR.
```

```
      PROCEDURE PRODXR(P,DEGREE,X)
         INTEGER VALUE DEGREE
         LONG ALGEBRAIC VALUE P,X

      #  THIS PROCEDURE MULTIPLIES A POLYNOMIAL P OF GIVEN
      #    DEGREE IN X BY X**R WHERE R IS AN INDETERMINATE.
      #  EXTERNAL VARIABLES REQUIRED: XXR.
      #  SYSTEM PROCEDURES REQUIRED: GETBLK.
      #  DECLARATION IN CALLING PROCEDURE: LONG ALGEBRAIC ALTRAN PRODXR.

         INTEGER J
         LONG ALGEBRAIC NEWP,COEF
         EXTERNAL LONG ALGEBRAIC ARRAY XXR

         NEWP = 0
         DO J = 0,DEGREE
            COEF =GETBLK(P,X,J)
            NEWP = NEWP + COEF*XXR(J)
         DOEND
         RETURN(NEWP)

   END   #  END OF PROCEDURE PRODXR.
```

```
PROCEDURE PKINIT(DEFLT)
   INTEGER ARRAY DEFLT

#  THIS PROCEDURE INITIALIZES THE PACKAGE FOR THE VARIOUS SOLVER
#  SOLVER MODULES.  THE VALUES OF THE PACKAGE PARAMETERS THAT MAY
#  BE SET BY THE USER ARE CHECKED AGAINST THE DEFAULT (MAXIMUM) VALUES
#  AND ARE RESET WHERE NECESSARY.
#  INPUT PARAMETER:
#    DEFLT      -  AN ARRAY OF DEFAULT (MAXIMUM) VALUES OF PARAMETERS
#                  INTERPRETED AS FOLLOWS:
#    DEFLT(1)   -  MAXORD, THE MAXIMUM ORDER OF ODE ALLOWED;
#    DEFLT(2)   -  MAXDEG, THE MAXIMUM DEGREE OF APPROXIMATION ALLOWED;
#    DEFLT(3)   -  MAXMU, THE MAXIMUM NUMBER OF INDERTERMINATES ALLOWED
#                  IN A PROBLEM;
#    DEFLT(4)   -  YDEG, THE MAXIMUM DEGREE OF ODE ALLOWED;
#    DEFLT(5)   -  MAXCOD, THE MAXIMUM VALUE OF THE SOLUTION CODE.
#  OTHER PACKAGE PARAMETERS:
#    MAXVAL     -  MAXIMUM NUMBER OF X-VALUES ALLOWED IN THE ASSOCIATED
#                  CONDITIONS;
#    XDEG       -  MAXIMUM EXPONENT OF THE INDEPENDENT VARIABLE X;
#    MAXPAR     -  MAXIMUM NUMBER OF SYMBOLIC PARAMETERS INTRODUCED
#                  DURING THE COMPUTATION;
#    MSGLVL     -  THE MESSAGE LEVEL INDICATOR, CONTROLLING THE
#                  AMOUNT OF INFORMATION PRINTED BY THE PACKAGE.
#   EXTERNAL VARIABLES:  ALL PACKAGE PARAMETERS ARE EXTERNAL VARIABLES.
#   SYSTEM PROCEDURES REQUIRED: IMIN,IMAX,NULL.
#   DECLARATION IN CALLING PROCEDURE: NONE.


   EXTERNAL INTEGER MAXORD,MAXDEG,MAXVAL,MAXCOD,MAXMU
   EXTERNAL INTEGER MAXPAR,XDEG,YDEG,MSGLVL,IERR,INPFLG

#  CHECK AND SET (INTERFACE) PACKAGE PARAMETERS.

   IF (NULL(MAXORD)) MAXORD = DEFLT(1)     #  MAXIMUM ORDER.
   ELSE IF (MAXORD<0 .OR. MAXORD>5) DO
        WRITE " *** ERROR 5011 *** - INVALID MAXIMUM ORDER",MAXORD
        MAXORD = DEFLT(1)
        WRITE " MAXIMUM ORDER RESET TO " , MAXORD
   DOEND
   ELSE MAXORD = IMAX(MAXORD,1)

   IF (NULL(MAXDEG)) MAXDEG = DEFLT(2)    #    MAXIMUM DEGREE.
   ELSE IF( MAXDEG<0 .OR. MAXDEG>15) DO
        WRITE " *** ERROR 5012 *** - INVALID MAXIMUM DEGREE",MAXDEG
        MAXDEG =  DEFLT(2)
        WRITE " MAXDEG RESET TO ", MAXDEG
   DOEND
```

```
      IF (NULL(MAXMU)) MAXMU = DEFLT(3)      #   NUMBER OF INDETERMINATES
      ELSE IF (MAXMU<0 .OR. MAXMU>DEFLT(3)) DO
           WRITE " ERROR 5013 ***"
           WRITE " INVALID MAXIMUM NUMBER OF INDETERMINATES"
           MAXMU = DEFLT(3)
           WRITE " MAXMU RESET TO ", MAXMU
      DOEND

      IF (NULL(MAXVAL)) MAXVAL = 5          #   NUMBER OF X-VALUES.
      ELSE IF (MAXVAL<0 .OR. MAXVAL> 5) DO
           WRITE " *** ERROR 5014 ***"
           WRITE " INVALID MAXIMUM NUMBER OF X-VALUES"
           MAXVAL = 5
           WRITE " MAXVAL RESET TO ", MAXVAL
      DOEND
      ELSE MAXVAL = IMAX(MAXVAL,1)

      IF (NULL(MSGLVL)) MSGLVL = 1          # MESSAGE INDICATOR.
      ELSE IF (MSGLVL <1 .OR. MSGLVL >4) DO
           WRITE " *** ERROR 5015 ***"
           WRITE " INVALID MESSAGE INDICATOR"
           MSGLVL = 1
           WRITE " MSGLVL RESET TO", MSGLVL
      DOEND
      WRITE MAXORD,MAXDEG,MAXMU,MAXVAL,MSGLVL

#     SET OTHER PACKAGE PARAMETERS.

      YDEG = DEFLT(4)
      MAXPAR = IMIN(15,MAXDEG+10)
      XDEG = IMIN(MAXDEG*YDEG + MAXPAR,40)
      MAXCOD = DEFLT(5)

      WRITE " "
      WRITE " ******* PACKAGE INITIALIZATION COMPLETED  ******* "
      WRITE " "

      RETURN
END #   END OF PROCEDURE PKINIT.
```

```
PROCEDURE PROLGL(N,ORDER,DIFFEQ,CODE,POLY,
                 RHS,MAXSUB,XSUB,CONDN,RANGE)
  INTEGER N,ORDER,MAXSUB,CODE
  RATIONAL ARRAY XSUB,RANGE
  LONG ALGEBRAIC DIFFEQ,RHS
  LONG ALGEBRAIC ARRAY POLY,CONDN

# THIS PROCEDURE READS IN AN ODE AND ITS ASSOCIATED CONDITIONS AND
# CHECKS THE VALIDITY OF THE ORDER, THE DEGREE OF APPROXIMATION
# REQUIRED, THE SOLUTION CODE, AND THE RANGE IN WHICH THE SOLUTION
# IS REQUIRED.  A FATAL ERROR OCCURS IF THE ODE IS NONLINEAR.
# THE ACTUAL INPUT AND PARAMETER CHECKING ARE PERFORMED BY
# A CALL TO PROCEDURE READER.
# INPUT PARAMETERS: NONE.
# OUTPUT PARAMTERS:
#    N      - DEGREE OF APPROXIMATION REQUIRED;
#    ORDER  - ORDER OF ODE;
#    MAXSUB - NUMBER OF X-VALUES IN ASSOCIATED CONDITIONS;
#    CODE   - SOLUTION CODE;
#    XSUB   - AN ARRAY XSUB(1),..,XSUB(MAXSUB) OF X-VALUES APPEARING
#             IN THE ASSOCIATED CONDITIONS;
#    RANGE  - A RATIONAL ARRAY (A,B) INDICATING THE FINITE INTERVAL
#             IN WHICH THE SOLUTION IS DESIRED;
#    DIFFEQ - THE ODE EXPRESSED AS DIFFEQ=0;
#    POLY   - ARRAY POLY(0),...,POLY(ORDER) OF POLYNOMIAL COEFFICIENTS
#             IN ODE;
#    RHS    - RIGHT-HAND-SIDE POLYNOMIAL OF ODE;
#    CONDN  - ARRAY CONDN(1),..,CONDN(ORDER) OF ASSOCIATED CONDITIONS.
# EXTERNAL VARIABLES REQUIRED: MSGLVL.
# PACKAGE PROCEDURES REQUIRED: READER, SETP1.
# DECLARATION IN CALLING PROCEDURE: NONE.

  LOGICAL LINEAR
  EXTERNAL INTEGER MSGLVL
  LONG ALGEBRAIC ARRAY ALTRAN SETP1

# READ IN AND CHECK ODE AND ASSOCIATED CONDITIONS.
  READER(N,ORDER,DIFFEQ,CODE,MAXSUB,XSUB,CONDN,RANGE,LINEAR)

# CHECK THAT ODE IS LINEAR.

  IF (.NOT. LINEAR) DO
    WRITE " *** ERROR 5024 ***"
    WRITE " INVALID ODE - ODE IS NOT LINEAR"
    FRETURN
  DOEND

# SET UP POLYNOMIAL COEFFICIENTS IN ODE.

   POLY = SETP1(ORDER,DIFFEQ,RHS)
   IF (MSGLVL > 1) DO
     WRITE " THE POLYNOMIAL COEFFICIENTS OF THE ODE ARE:",POLY
     WRITE "THE RIGHT HAND SIDE POLYNOMIAL IS:", RHS
   DOEND

   RETURN
END # END OF PROCEDURE PROLGL.
```

```
PROCEDURE QROUND(Q,K,M)
   INTEGER VALUE K,M
   LONG RATIONAL ARRAY VALUE Q

#  THIS PROCEDURE ROUNDS THE ELEMENTS OF THE RATIONAL ARRAY
#    Q(0), .... , Q(K) TO M DECIMAL DIGITS.
#  PACKAGE PROCEDURES REQUIRED: LQABS.
#  SYSTEM PROCEDURES REQUIRED: INUM,IFLR.
#  DECLARATION IN CALLING PROCEDURE: LONG RATIONAL ARRAY ALTRAN QROUND.

   INTEGER I
   LONG INTEGER A,B,C,DEN,REM
   LONG RATIONAL R
   LONG INTEGER ALTRAN IFLR=S9IFLR
   LONG RATIONAL ALTRAN LQABS

   DEN = 10**M
   DO I = 0,K
      R = Q(I)
      A = INUM(R,B)

      C = IFLR(A*DEN,B,REM)
      IF ( (2*REM) > B ) C = C + 1
      Q(I) = C/DEN
    DOEND

    RETURN(Q)

END #  END OF PROCEDURE QROUND.
```

```
   INTEGER NMAX,ORDER,CODE,MAXSUB
   LOGICAL LINEAR
   RATIONAL ARRAY XSUB,RANGE
   LONG ALGEBRAIC DIFFEQ
   LONG ALGEBRAIC ARRAY CONDN

#   THIS PROCEDURE READS IN AND CHECKS AN ODE PROBLEM, AND THE
#     ASSOCIATED PROBLEM PARAMETERS.
#  EXTERNAL VARIABLES REQUIRED: INPFLG,MAXORD,MAXDEG,MAXCOD.
#  PACKAGE PROCEDURES REQUIRED: INPODE,INPCON,TRANS1.
#  DECLARATION IN CALLING PROCEDURE: NONE.

   RATIONAL ARRAY(2) RG
   EXTERNAL INTEGER INPFLG,MAXORD,MAXDEG,MAXCOD

#  READ IN ODE AND ASSOCIATED CONDITIONS.

   INPFLG = 0
   READ ORDER,NMAX,RG,CODE
   RANGE = RG
   WRITE ORDER,NMAX,RANGE,CODE
   INPODE(ORDER,DIFFEQ,LINEAR)
   INPCON(ORDER,MAXSUB,XSUB,CONDN)
   INPFLG = 1

#  CHECK PROBLEM PARAMETERS.

   IF (ORDER<0 .OR. ORDER>MAXORD) DO          # ORDER.
      WRITE " *** ERROR 5021 *** "
      WRITE " ORDER OUT OF RANGE "
      FRETURN
   DOEND

   IF (NMAX<0 .OR. NMAX>MAXDEG) DO            # NMAX.
      WRITE " *** ERROR 5022 *** "
      WRITE " NMAX OUT OF RANGE "
      FRETURN
   DOEND

   IF (CODE<1 .OR. CODE>MAXCOD) DO            # CODE.
      WRITE " *** ERROR 5023 *** "
      WRITE " SOLUTION CODE OUT OF RANGE"
      FRETURN
   DOEND

   IF (RANGE(1) >= RANGE(2)) DO               # RANGE.
      WRITE " *** ERROR 5027 *** "
      WRITE " INVALID RANGE"
      FRETURN
   DOEND

   TRANSFORM PROBLEM TO (-1,1) IF NECESSARY.

   IF (.NOT. RANGE==(-1,1)) DO
      TRANS1(ORDER,RANGE,DIFFEQ,CONDN,XSUB,MAXSUB)
      WRITE " PROBLEM TRANSFORMED TO RANGE (-1,1)"
      WRITE " THE TRANSFORMED PROBLEM IS GIVEN BY"
      WRITE DIFFEQ,CONDN,XSUB
   DOEND

   RETURN
END # END OF PROCEDURE READPR
```

```
PROCEDURE REDUZS(Q,Z,Z1,G,G1,H,N,GENPOL,V,V1,S,S1)
    INTEGER VALUE G1,H,N,Z1
    INTEGER V1=0,S1=Z1
    INTEGER ARRAY VALUE G,Z
    INTEGER ARRAY S=Z
    LONG ALGEBRAIC VALUE GENPOL
    LONG ALGEBRAIC ARRAY (0:N+H) Q
    LONG ALGEBRAIC ARRAY V

#  THIS PROCEDURE OBTAINS THE MINIMAL SET S OF INDICES OF
#    UNDEFINED CANONICAL POLYNOMIALS ASSOCIATED WITH A
#    LINEAR OPERATOR DEFINED BY THE GENERATING POLYNOMIAL
#    GENPOL.
#  EXTERNAL VARIABLES REQUIRED: XX,XXR,XUQ.
#  PACKAGE PROCEDURES REQUIRED: SUBS.
#  SYSTEM PROCEDURES REQUIRED: TPS,GETBLK,DEG.
#  DECLARATION IN CALLING PROCEDURE: NONE.

    INTEGER I,J,K,L,M,RP
    LONG ALGEBRAIC ARRAY (0:H) COEF
    LONG ALGEBRAIC B,C,JP,T,SUM
    LONG ALGEBRAIC ARRAY XLIST,VALIST,L1,L2,A
    EXTERNAL LONG ALGEBRAIC XX
    EXTERNAL LONG ALGEBRAIC ARRAY XXR,XUQ
    LONG ALGEBRAIC ALTRAN SUBS
    LONG ALGEBRAIC ARRAY ALTRAN TPS

#  COMMENCE REDUCTION IF POSSIBLE.

    IF (G1 < 1) RETURN  #  NO REDUCTION POSSIBLE.

    DO I = 0,H
      COEF(I) = GETBLK(GENPOL,XXR(I),1)
    DOEND
    XLIST=0$0; VALIST=0$0
    L1 = 0$0; L2 =Z1$0
    DO I = 1,Z1
      L1 = (L1,XUQ(Z(I)))
    DOEND
    V1 = 0;  L=1;  M=0

BACK:  IF ( L > G1) GO TO FINI
    K = G(L)
    C = XX**K
    B = SUBS(C,COEF,H,XX)
    L = L+1
```

```
CHECKB:
    IF (B==0) DO        #  IS C A POLYNOMIAL SOLUTION?
       V = (V,C)
       V1 = V1 + 1
       GO TO BACK
    DOEND

    ELSE DO
       RP = DEG(B,XX)
       A = TPS(B,XX,RP)
       JP = A(RP)
       IF ( Q(RP)<>XUQ(RP) ) DO
          T = Q(RP)(L1=L2)
          B = B - JP*SUBS(T,COEF,H,XX)
          C = C - JP*T
          GO TO CHECKB
       DOEND

       ELSE DO   #  DEFINE Q(RP).
          SUM = C
          DO I = 0,RP-1
             SUM = SUM - Q(I)*A(I)
          DOEND

          Q(RP) = SUM/JP
          M = M+1
          XLIST = (XLIST,XUQ(RP))
          VALIST = (VALIST,Q(RP))
          GO TO BACK
       DOEND
    DOEND
FINI:  #  SUBSTITUTE FOR ANY NEWLY DEFINED CAN. POLYS.
    IF (M>0) DO
       S1 = 0; S=0$0
       DO I =0,N+H
          IF (Q(I) == XUQ(I)) DO
             S1 = S1 + 1
             S = (S,I)
          DOEND
          ELSE Q(I) = Q(I)(XLIST=VALIST)
       DOEND
    DOEND

END #  END OF PROCEDURE REDUZS
```

```
PROCEDURE SETP1(ORDER,DIFFEQ,RHS)
   INTEGER VALUE ORDER
   LONG ALGEBRAIC VALUE DIFFEQ
   LONG ALGEBRAIC RHS

#  THIS PROCEUDRE RETURNS THE POLYNOMIAL COEFFICIENTS OF
#    A LINEAR ODE OF GIVEN ORDER,  THE RIGHT-HAND-SIDE
#    POLYNOMIALS ARE RETURNED IN RHS.
#  EXTERNAL VARIABLES REQUIRED: MSGLVL,XX,XY,XDY.
#  SYSTEM PROCEDURES REQUIRED: GETBLK.
#  DECLARATION IN CALLING PROCEDURE: LONG ALGEBRAIC ARRAY ALTRAN SETP1.

   INTEGER I,J,K,NEWORD
   LONG ALGEBRAIC P,LHS
   LONG ALGEBRAIC ARRAY (0:ORDER) POLY=0
   EXTERNAL INTEGER MSGLVL
   EXTERNAL LONG ALGEBRAIC XX,XY
   EXTERNAL LONG ALGEBRAIC ARRAY XDY

   IF (MSGLVL>3) WRITE "ENTER SETP1"
#  PICK OFF THE POLYNOMIAL COEFFICIENTS.

    LHS = 0
    NEWORD = -1
    DIFFEQ = DIFFEQ(XY=XDY(0))

    DO I = 0,ORDER
       P = GETBLK(DIFFEQ,XDY(I),1)
       IF (P<>0) DO
          POLY(I) = P
          NEWORD = I
          LHS = LHS + P*XDY(I)
       DOEND
    DOEND
    RHS = LHS - DIFFEQ

    IF (NEWORD <> ORDER) DO
       WRITE "  *** ERROR 5025 *** "
       WRITE "  WRONG ORDER SPECIFIED FOR ODE", ORDER
       WRITE "  ODE IS ACTUALLY OF ORDER", NEWORD
       FRETURN
    DOEND

   IF (MSGLVL>3) WRITE POLY,RHS, "EXIT SETP1"
   RETURN(POLY)

END   #  END OF PROCEDURE SETP1.
```

```
PROCEDURE SETP2(N,ORDER,MAXN,DIFFEQ,YINIT,RHS)
   INTEGER VALUE N,ORDER,MAXN
   LONG ALGEBRAIC VALUE DIFFEQ,YINIT
   LONG ALGEBRAIC RHS

#  THIS PROCEDURE RETURNS THE POLYNOMIAL COEFFICIENTS,POLY,
#    OF THE LINEAR ODE OBTAINED FROM A NONLINEAR ODE (OF GIVEN
#    ORDER), DIFFEQ, BY NEWTON ITERATION. THE
#    RIGHT-HAND-SIDE POLYNOMIAL OF THE LINEAR ODE IS
#    RETURNED IN RHS.
#    YINIT IS THE CURRENT ESTIMATE FOR THE SOLUTION FUNCTION Y(X).
#    THE POLYNOMIAL COEFFICIENTS ARE TRUNCATED TO DEGREE MAXN IN X.
#  EXTERNAL VARIABLES REQUIRED:  MSGLVL,XX,XDY,XY.
#  PACKAGE PROCEDURES REQUIRED: TRUNC.
#  SYSTEM PROCEDURES REQUIRED: DIFF.
#  DECLARATION IN CALLING PROCEDURE: LONG ALGEBRAIC ARRAY ALTRAN SETP2.

   INTEGER I,J,K,NEWORD
   LONG ALGEBRAIC P,LHS
   LONG ALGEBRAIC ARRAY YLIST,YVAL
   LONG ALGEBRAIC ARRAY(0:ORDER) POLY=0
   EXTERNAL INTEGER MSGLVL
   EXTERNAL LONG ALGEBRAIC XX,XY
   EXTERNAL LONG ALGEBRAIC ARRAY XDY
   LONG ALGEBRAIC ALTRAN TRUNC

   IF (MSGLVL>3) WRITE "ENTER SETP2"
   LHS = 0
   NEWORD = -1
   DIFFEQ = DIFFEQ(XY=XDY(0))

   SET UP LISTS OF Y AND DERIVATIVE VALUES.

   YLIST = XDY(0)
   YVAL = YINIT
   P = YINIT
   DO I = 1,ORDER
      P = DIFF(P,XX)
      YLIST = (YLIST,XDY(I))
      YVAL = (YVAL,P)
   DOEND
```

```
#  NOW GET THE COEFFICIENTS.

   RHS = DIFFEQ(YLIST = YVAL)
   RHS = - TRUNC(RHS,N,XX)
   DO I =0,ORDER
      P = DIFF(DIFFEQ,XDY(I))(YLIST=YVAL)
      IF (P<>0) DO
         P = TRUNC(P,MAXN,XX)
         NEWORD = I
         POLY(I) = P
      DOEND
   DOEND

#  CHECK THE ORDER OF ODE.

   IF (NEWORD <> ORDER) DO
      WRITE " *** ERROR 5025 ***"
      WRITE " WRONG ORDER SPECIFIED FOR ODE",ORDER
      WRITE " ODE IS ACTUALLY OF ORDER",NEWORD
      FRETURN
    DOEND

   IF (MSGLVL>3) WRITE POLY,RHS,"EXIT SETP1"
   RETURN(POLY)
END  #  END OF PROCEDURE SETP2.
```

List of Level 6 Procedures

List of Level 6 Procedures

```
PROCEDURE ABS(X)
  LONG REAL X

#  THIS PROCEDURE RETURNS THE ABSOLUTE VALUE OF
#  THE REAL NUMBER X.
#  DECLARATION IN CALLING PROCEDURE: REAL ALTRAN ABS.

  IF (X < 0.0) RETURN(-X)
  ELSE RETURN(X)

END #  END OF PROCEDURE ABS.
```

```
PROCEDURE AMAX(P,DEGREE,X)
   INTEGER VALUE DEGREE
   LONG ALGEBRAIC VALUE P,X

#  THIS PROCEDURE RETURNS THE MAXIMUM ABSOLUTE VALUE
#    OF A POLYNOMIAL P, OF GIVEN DEGREE IN X, IN THE
#    INTERVAL -1<=X<=1.  THE COEFFICIENTS OF P ARE
#    ASSUMED TO BE RATIONAL NUMBERS.
#  PACKAGE PROCEDURES REQUIRED: LQABS,CHFORM,TCS
#  DECLARATION IN CALLING PROCEDURE: LONG ALGEBRAIC ALTRAN AMAX.

   INTEGER I
   LONG ALGEBRAIC MAXP, PMAX
   LONG ALGEBRAIC ARRAY A
   LONG RATIONAL ALTRAN LQABS
   LONG ALGEBRAIC ALTRAN CHFORM
   LONG ALGEBRAIC ARRAY ALTRAN TCS

   PMAX = CHFORM(P,DEGREE,X)
   A = TCS(PMAX,DEGREE)

   MAXP = LQABS(A(0))/2
   DO I = 1,DEGREE
      MAXP = MAXP + LQABS(A(I))
   DOEND

   RETURN(MAXP)

END # END OF PROCEDURE AMAX.
```

```
PROCEDURE CHFORM(P, DEGREE, X)
  INTEGER VALUE  DEGREE
  LONG ALGEBRAIC VALUE  P, X

  # PROCEDURE TO COMPUTE THE CHEBYSHEV FORM OF A POLYNOMIAL P
  #    OF GIVEN DEGREE IN THE INDETERMINATE X.  IT IS ASSUMED
  #    THAT THE EXTERNAL ARRAY XPOWER HAS BEEN INITIALIZED TO
  #    CONTAIN THE CHEBYSHEV FORM OF THE POWER X**K AS ITS
  #    ELEMENT XPOWER(K), FOR 0 <= K <= DEGREE.
  # EXTERNAL VARIABLES REQUIRED: XPOWER.
  # SYSTEM PROCEDURES REQUIRED: GETBLK.
  # DECLARATION IN CALLING PROCEDURE: LONG ALGEBRAIC ALTRAN CHFORM.


  INTEGER  K
  LONG ALGEBRAIC  COEF, NEWP
  EXTERNAL  LONG ALGEBRAIC ARRAY  XPOWER

  NEWP = 0
  DO K = 0, DEGREE
    COEF = GETBLK(P, X, K)
    NEWP = NEWP + COEF*XPOWER(K)
  DOEND

  RETURN(NEWP)

END  # END OF PROCEDURE CHFORM.
```

```
PROCEDURE COM(K)
   INTEGER VALUE K
   INTEGER I,J
   INTEGER ARRAY (1:K,1:K) COMB

#  THIS PROCEDURE RETURNS A K*K ARRAY OF BINOMIAL COEFFICIENTS.
#  DECLARATION IN CALLING PROCEDURE: INTEGER ARRAY ALTRAN COM.

   DO I = 1,K
      COMB(I,1) = I
      DO J = 2,I-1
         COMB(I,J) = COMB(I-1,J-1) + COMB(I-1,J)
      DOEND
      COMB(I,I) = 1
   DOEND

   RETURN(COMB)
END # END OF PROCEDURE COM.
```

```
PROCEDURE CWRIT(C,N)
   INTEGER VALUE N
   LONG ALGEBRAIC ARRAY VALUE C

#  THIS PROCEDURE PRINTS THE N+1 COEFFICIENTS CHEBYSHEV COEFFICIENTS
#  AS REALS IF POSSIBLE (I.E. IF THE COEFFICIENTS CONTAIN NO
#  INDETERMINATES)  AND OTHERWISE IN SYMBOILC FORM.
#  DECLARATION IN CALLING PROCEDURE: NONE.

   INTEGER I,J
   LONG REAL ARRAY(0:N) CIREAL
   RATIONAL ALTRAN LQT

   WRITE "THE CHEBYSHEV COEFFICIENTS ARE:"

   DO J = 0,N
      C(J) = LQT(C(J),ASIS)
      CIREAL(J) = LR(C(J))
      WRITE CIREAL(J)
      GO TO ON
ASIS: WRITE C(J)
ON: DOEND

   RETURN

END #  END OF PROCEDURE CWRIT.
```

```
PROCEDURE EVAL(C,DEGREE,X)
   INTEGER VALUE DEGREE
   LONG ALGEBRAIC VALUE X
   LONG ALGEBRAIC ARRAY VALUE C

#  THIS PROCEDURE EVALUATES A TRUNCATED CHEBYSHEV SERIES C
#  OF GIVEN DEGREE AT THE GIVEN ALGEBRAIC POINT X.
#  DECLARATION IN CALLING PROCEDURE: LONG ALGEBRAIC ALTRAN EVAL.

   INTEGER I
   LONG ALGEBRAIC BOLD,TEMP,TWOX,B

   TWOX = 2*X
   BOLD = 0; TEMP = 0
   B = C(DEGREE)

   DO I = DEGREE-1,0,-1
      BOLD = TEMP
      TEMP = B
      B = TWOX*B - BOLD + C(I)
   DOEND

   RETURN( (B-BOLD)/2)

END # END OF PROCEDURE EVAL.
```

```
PROCEDURE IABS(K)
   INTEGER VALUE K

#  THIS PROCEDURE RETURNS THE ABSOLUTE VALUE OF THE SHORT
#    INTEGER K.
#  DECLARATION IN CALLING PROCEDURE: INTEGER ALTRAN IABS.

   IF (K<0) RETURN(-K)
   ELSE RETURN(K)

END  # END OF PROCEDURE IABS.
```

```
 PROCEDURE INFORM(ORDER,POLY, X)
INTEGER VALUE ORDER
LONG ALGEBRAIC VALUE X
LONG ALGEBRAIC ARRAY VALUE POLY

 #  THIS PROCEDURE CONVERTS A LINEAR ODE WITH POLYNOMIAL
 #    COEFFICIENTS TO INTEGRATED FORM.
 #    THE COEFFICIENTS OF THE INTEGRATED FORM ARE RETURNED
 #    IN Q(0),...,Q(K), WHERE K IS THE ORDER OF THE ODE.
 #  PACKAGE PROCEDURES REQUIRED: COM
 #  SYSTEM PROCEDURES REQUIRED: DIFFN.
 #  DECLARATION IN CALLING PROCEDURE: LONG ALGEBRAIC ARRAY ALTRAN INFORM.

INTEGER M,I,SIGN
INTEGER ARRAY COMB
LONG ALGEBRAIC P
LONG ALGEBRAIC ARRAY(0:ORDER) Q
LONG ALGEBRAIC ARRAY XLIST
INTEGER ARRAY ALTRAN COM
LONG ALGEBRAIC ALTRAN DIFFN

COMB = COM(ORDER)
Q(0) = POLY(ORDER)
DO M = 1,ORDER
   #  DETERMINE THE M-TH POLYNOMIAL IN THE INTEGRATED FORM.
   P = POLY(ORDER-M) - (ORDER-M+1)*DIFFN(POLY(ORDER-M+1),X)
   SIGN = -1
   XLIST = 1$X
   DO I = M-2,0,-1
      SIGN = -SIGN
      XLIST = (XLIST,X)
      P = P + SIGN*COMB(ORDER-I,M-I)*DIFFN(POLY(ORDER-I),XLIST)
   DOEND
   Q(M) = P
DOEND

RETURN(Q)
END  #  END OF PROCEDURE INFORM.
```

```
PROCEDURE LQABS(K)
  LONG RATIONAL VALUE K

#  THIS PROCEDURE RETURNS THE ABSOLUTE VALUE OF THE LONG
#     RATIONAL NUMBER K.
#  DECLARATION IN CALLING PROCEDURE: LONG RATIONAL ALTRAN LQABS.

 IF (K>=0) RETURN(K)
 ELSE RETURN(-K)

END  #  END OF PROCEDURE LQABS.
```

```
PROCEDURE POWERS(MAXDEG)
  INTEGER VALUE  MAXDEG

  # PROCEDURE TO COMPUTE THE CHEBYSHEV FORM OF THE POWERS X**K
  #    FOR 0 <= K <= MAXDEG.  THE K-TH CHEBYSHEV POLYNOMIAL IS
  #    REPRESENTED BY THE INDETERMINATE XT(K), WHERE XT IS AN
  #    EXTERNAL ALGEBRAIC ARRAY OF INDETERMINATES.  THE VALUE
  #    RETURNED IS AN ARRAY DIMENSIONED 0 TO MAXDEG CONTAINING
  #    THE APPROPRIATE CHEBYSHEV FORMS.
  #  EXTERNAL VARIABLES REQUIRED: XT.
  #  SYSTEM PROCEDURES REQUIRED: IMOD.
  #  DECLARATION IN CALLING PROCEDURE: LONG ALGEBRAIC ARRAY ALTRAN POWERS.



  INTEGER  K, HALFK, KMOD2, J
  LONG INTEGER ARRAY  KCHOOS, TEMP
  LONG ALGEBRAIC ARRAY (0:MAXDEG)  A
  EXTERNAL LONG ALGEBRAIC ARRAY  XT


  A(0) = XT(0)
  IF (MAXDEG > 0)  A(1) = XT(1)
  IF (MAXDEG > 1)  A(2) = (XT(2)+XT(0))/2
  KCHOOS = 1$(1)

  DO K = 3, MAXDEG

     KMOD2 = IMOD(K, 2, HALFK)

     # UPDATE ARRAY KCHOOS SO THAT KCHOOS(J) = "K CHOOSE J",
     #     FOR J = 1, ..., K/2 .

     IF (KMOD2 == 1)  TEMP = KCHOOS
        ELSE  TEMP = (KCHOOS, KCHOOS(HALFK-1))
     KCHOOS = HALFK$(0)  # CREATES ARRAY OF DESIRED LENGTH.
     KCHOOS(1) = K
     DO J = 2, HALFK
        KCHOOS(J) = TEMP(J) + TEMP(J-1)
     DOEND


     # COMPUTE CHEBYSHEV FORM OF X**K INTO ARRAY ELEMENT A(K).

     A(K) = XT(K)
     DO J = 1, HALFK-1
        A(K) = A(K)  +  KCHOOS(J) * XT(K - 2*J)
     DOEND
     IF (KMOD2==1) A(K) = (A(K) + KCHOOS(HALFK)*XT(1))/2**(K-1)
        ELSE  A(K) = (A(K) + 1/2*KCHOOS(HALFK)*XT(0))/2**(K-1)
  DOEND

  RETURN( A )

END  # END OF PROCEDURE POWERS #
```

```
PROCEDURE PREC(X)
   REAL VALUE X

#  THIS PROCEDURE COMPUTES THE PRECISION OF THE REAL NUMBER,X,
#    THAT IS, IT RETURNS THE INTEGER M SUCH THAT 10**(-M) <= X.
#  DECLARATION IN CALLING PROCEDURE: INTEGER ALTRAN PREC.

   INTEGER M
   REAL Y,T

   M = 0
   Y = 1.0
   T = 0.1
 BACK:
   IF (Y>X) DO
      M = M + 1
      Y = Y*T
      GO TO BACK
   DOEND

   RETURN(M)

END # END OF PROCEDURE PREC.
```

104

```
PROCEDURE SUBS(P,COEF,H,X)
    INTEGER VALUE H
    LONG ALGEBRAIC VALUE P,X
    LONG ALGEBRAIC ARRAY VALUE COEF

#  THIS PROCEDURE SUBSTITUTES A POLYNOMIAL P INTO
#    A GENERATING POLYNOMIAL (OF HEIGHT H), WHOSE COEFFICIENTS
#    ARE GIVEN IN ARRAY COEF.
#  EXTERNAL VARIABLES REQUIRED: RR.
#  SYSTEM PROCEDURES REQUIRED: TPS, DEG.
#  DECLARATION IN CALLING PROCEDURE: LONG ALGEBRAIC ALTRAN SUBS.

    INTEGER I,J,DEGREE
    LONG ALGEBRAIC SUM,T,Q
    LONG ALGEBRAIC ARRAY B
    EXTERNAL LONG ALGEBRAIC RR
    LONG ALGEBRAIC ARRAY ALTRAN TPS

    DEGREE = DEG(P,X)
    B = TPS(P,X,DEGREE)
    Q = 0
    DO I = 0,DEGREE
       SUM = 0
       DO J = 0,H
          T = COEF(J)(RR=I)
          IF (T<> 0) SUM = SUM + T*X**(I+J)
        DOEND

        Q = Q  + B(I)*SUM
      DOEND

     RETURN(Q)
END  #  END OF PROCEDURE SUBS.
```

```
PROCEDURE TCS(P,DEGREE)
   INTEGER VALUE DEGREE
   LONG ALGEBRAIC VALUE P

#  THIS PROCEDURE EXTRACTS THE CHEBYSHEV COEFFICIENTS, C,
#    OF A POLYNOMIAL P WHICH IS ASSUMED TO BE IN THE FORM
#     C(0)*XT(0)/2 + C(1)*XT(1) + .... + C(N)*XT(N),
#    WHERE N=DEGREE, AND XT(I) REPRESENTS THE I-TH CHEBYSHEV
#    POLYNOMIAL.
#  EXTERNAL VARIABLES REQUIRED: XT.
#  SYSTEM PROCEDURES REQUIRED: GETBLK.
#  DECLARATION IN CALLING PROCEDURE: LONG ALGEBRAIC ARRAY ALTRAN TCS.

   INTEGER K
   LONG ALGEBRAIC ARRAY(0:DEGREE) C
   EXTERNAL LONG ALGEBRAIC ARRAY XT

   DO K = 0,DEGREE
      C(K) = GETBLK(P,XT(K),1)
      P = P - C(K)*XT(K)
   DOEND
   C(0) = C(0)*2

   RETURN(C)
END  #  END OF PROCEDURE TCS.
```

```
PROCEDURE TPOWS(MAXDEG,X)
   INTEGER VALUE MAXDEG
   LONG ALGEBRAIC VALUE X

#  THIS PROCEDURE RETURNS THE CHEBYSHEV POLNOMIALS OF DEGREE
#    NOT EXCEEDING MAXDEG.  THE POLNOMIALS ARE IN THE
#    INDEPENDENT VARIABLE X.
#  DECLARATION IN CALLING PROCEDURE: LONG ALGEBRAIC ARRAY ALTRAN TPOWS.

   LONG ALGEBRAIC ARRAY (0:MAXDEG) A
   LONG ALGEBRAIC  R = 1, S,Y, T
   INTEGER I

   Y = X
   S = Y
   A(0) = R
   IF (MAXDEG == 0) RETURN(A)
   A(1) = S
   IF (MAXDEG == 1) RETURN(A)

   DO I = 2,MAXDEG
      T = 2*Y*S - R
      A(I) = T
      R = S
      S = T
   DOEND

 RETURN(A)
END # END OF PROCEDURE TPOWS.
```

```
PROCEDURE TRANS1(ORDER,RANGE,DIFFEQ,CONDN,XSUB,MAXSUB)
   INTEGER VALUE ORDER,MAXSUB
   RATIONAL ARRAY RANGE,XSUB
   LONG ALGEBRAIC DIFFEQ
   LONG ALGEBRAIC ARRAY CONDN

#  THIS PROCEDURE TRANSFORMS AN ODE (AND ASSOCIATED
#    CONDITIONS) FROM AN ARBITRARY FINITE RANGE TO THE
#    INTERVAL (-1,1).
#  EXTERNAL VARIABLES REQUIRED: XX,XDY,XDYX.
#  DECLARATION IN CALLING PROCEDURE: NONE.

   INTEGER I,J
   RATIONAL A,B
   LONG ALGEBRAIC P,Q
   LONG ALGEBRAIC ARRAY UNLIST,VALIST
   EXTERNAL LONG ALGEBRAIC XX
   EXTERNAL LONG ALGEBRAIC ARRAY XDY,XDYX

   A = RANGE(1)
   B = RANGE(2)
   P = 2/(B-A)

#  PERFORM TRANSFORMATION ON DIFFEQ.

   UNLIST = (XX)
   VALIST = (((B-A)*XX + (B+A))/2)
   Q = P
   DO I = 1,ORDER
      UNLIST = (UNLIST,XDY(I))
      VALIST = (VALIST,Q*XDY(I))
      Q = Q*P
   DOEND
   DIFFEQ = DIFFEQ(UNLIST=VALIST)

#  PERFORM TRANSFORMATION ON ASSOCIATED CONDITIONS.

   UNLIST = 0$0;  VALIST = 0$0
   Q = P
   DO I =1,ORDER-1
      DO J = 1,MAXSUB
         UNLIST = (UNLIST,XDYX(I,J))
         VALIST = (VALIST,Q*XDYX(I,J))
      DOEND
      Q = Q*P
   DOEND

   DO I = 1,ORDER
      CONDN(I) = CONDN(I)(UNLIST = VALIST)
   DOEND
   Q = (2*XX - (B+A))/(B-A)
   DO I = 1,MAXSUB
      XSUB(I) = Q(XX=XSUB(I))
   DOEND
   RETURN
END  #  END OF PROCEDURE TRANS1
```

```
PROCEDURE TRANS2(RANGE,P,NEWP)
   RATIONAL ARRAY VALUE RANGE
   LONG ALGEBRAIC VALUE P
   LONG ALGEBRAIC NEWP

#  THIS PROCEDURE TRANSFORMS THE POLYNOMIAL P FROM
#    INTERVAL (-1,1) TO AN ARBITRARY (GIVEN) FINITE
#    RANGE, GIVING NEWP.
#  EXTERNAL VARIABLES REQUIRED: XX
#  DECLARATION IN CALLING PROCEDURE: NONE.

   RATIONAL A,B
   LONG ALGEBRAIC Z
   EXTERNAL LONG ALGEBRAIC XX

   A = RANGE(1)
   B = RANGE(2)
   Z = (2*XX - (B+A))/(B-A)

   NEWP = P(XX=Z)

    RETURN
END  # END OF PROCEDURE TRANS2.
```

```
109
    PROCEDURE TRUNC(P,DEGREE,X)
    INTEGER VALUE DEGREE
    LONG ALGEBRAIC VALUE P,X

    THIS PROCEDURE TRUNCATES A POLYNOMIAL P TO A GIVEN
      DEGREE IN X.
    PACKAGE PROCEDURES REQUIRED: TCS,CHFORM,EVAL.
    DECLARATION IN CALLING PROCEDURE: LONG ALGEBRAIC ALTRAN TRUNC.

    INTEGER DEGP
    LONG ALGEBRAIC ARRAY B
    LONG ALGEBRAIC ALTRAN EVAL,CHFORM
    LONG ALGEBRAIC ARRAY ALTRAN TCS

    DEGP = DEG(P,X)
    IF (DEGP<=DEGREE) RETURN(P)
    B = TCS(CHFORM(P,DEGP,X),DEGREE)
    P = EVAL(B,DEGREE,X)

    RETURN(P)
END; #  END OF PROCEDURE TRUNC.

PROCEDURE TPOWS(MAXDEG,X)
    INTEGER VALUE MAXDEG
    LONG ALGEBRAIC VALUE X

    THIS PROCEDURE RETURNS THE CHEBYSHEV POLNOMIALS OF DEGREE
      NOT EXCEEDING MAXDEG.  THE POLNOMIALS ARE IN THE
      INDEPENDENT VARIABLE X.
    DECLARATION IN CALLING PROCEDURE: LONG ALGEBRAIC ARRAY ALTRAN TPOWS;

    LONG ALGEBRAIC ARRAY (0:MAXDEG) A
    LONG ALGEBRAIC  R = 1, S,Y, T
    INTEGER I

    Y = X
    S = Y
    A(0) = R
    IF (MAXDEG == 0) RETURN(A)
    A(1) = S
    IF (MAXDEG == 1) RETURN(A)

    DO I = 2,MAXDEG
      T = 2*Y*S - R
      A(I) = T
      R = S
      S = T
    DOEND

  RETURN(A)
END; # END OF PROCEDURE TPOWS.
```

4. References

[1] Afolabi, M.O., "Symbolic Series Solution of Ordinary Differential Equations", Ph.D. Thesis, University of Waterloo, Waterloo, Ontario, Canada, 1979.

[2] Brown, W.S., ALTRAN User's Manual, Bell Laboratories, Murray Hill, New Jersey, 1977.

[3] Fox, P.A., A.D. Hall, and N.L. Schryer, "The PORT Mathematical Subroutine Library", TOMS, 4 (1978), pp. 104-126.

[4] George, A., and J.W.H. Liu, "The Design and Implementation of a Sparse Matrix Package", Research Report CS-77-21, Department of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, 1977.